Report - Machine Learning (HWS 2024)

Assignment 1: Naive Bayes

Arne Huckemann (ahuckema), Elise Wolf (eliwolf)

October 12, 2024

This report examines the Naive Bayes classification algorithm on the MNIST digit dataset, focusing on the effects of Laplace smoothing. We trained and evaluated the model using standard metrics, including accuracy, precision, recall, and F1-score. Cross-validation was used to determine the optimal smoothing parameter α, revealing its impact on model performance. We also generated digit samples using the learned distributions to study the influence of α on sample quality. Lastly, theoretical aspects, such as handling missing data, were discussed. The findings underscore the importance of tuning α to balance overfitting and underfitting in classification and generative tasks.

1. Introduction

To explain our solutions of the following tasks, we will introduce the following notation.

Definition 1.1. *We define our original state space $\mathfrak{X}' := \{0, ..., K-1\}^{d \times d}$ where $d \in \mathbb{N}$ denotes the dimension. Therefore, $X \in \mathfrak{X}$ denotes a picture of dimension $d \times d$. Our classes space is given by $\Theta := \{0, ..., C-1\}$. As $\mathfrak{X}'$ and Θ are discrete sets, the following $\sigma-$algebras are the respective power sets.*

Remark 1.1. *For the analysis of the MNIST digits dataset provided, the state space is given by $\mathfrak{X} = \{0, ..., 255\}^{28 \times 28}$ consisting of $K = \{0, ..., 255\}$ possible values of each feature. As the code given in the exercises suggested to view the pictures as vectors, we define $\mathfrak{X} := \{0, ..., K-1\}^{d \cdot d}$, where $d = 28$. Our class space $\Theta := \{0, ..., 9\}$ consists of the ten possible numbers the handwriting in picture X can refer to. The dataset consists of $N = 60000$ training examples and $N_{test} = 10000$ test examples. For further implementation and notation we will denote $X_{i,j,k}$ as sample $i \in \{1, ..., N\}$ of the dataset with pixel $j \in \{0, ..., 784\}$ taking a $k \in \{0, ..., 255\}$ value on the gray scale.*

Definition 1.2 (Bayesian Model)**.** *For a distribution Q consider the tupel $(\mathfrak{X}, \Theta, Q)$ and an implicitly underlying probability space $(\Omega, \mathcal{A}, Q)$. We call $(X, \theta) : (\Omega, \mathcal{A}) \to$*

$(\mathfrak{X} \times \Theta, \mathcal{P}(\mathfrak{X} \times \Theta))$ *a Bayesian Model. We call the marginal* Q^θ *of* $Q^{(X,\theta)}$ *on* Θ *prior denoted by*

$$\pi(A) := Q^\theta(A) = Q^{(X,\theta)}(\mathfrak{X} \times A), \quad A \in \mathcal{F}(A).$$

We call the conditional probability $Q^{\theta|X=x}$ *posterior.*

2. Task 1 and 2

A formal definition and solution for Task 1 and 2 can be found in the Appendix A

3. Task 3 - Experiments on MNIST Digits Data

In the following section we are going to compare the results of our trained models *model_nb* with $\alpha = 1$ and *model_nb2* with $\alpha = 2$ as a model with add-one-smoothing.

3.1. Part a) Evaluation of the Model with $\alpha = 2$

We trained our Naive Bayes model with $\alpha = 2$ on the MNIST training dataset and evaluated it on the test set. The complete evaluation of those metrics can be found in Table 1 for our model trained with $\alpha = 2$. Additional, to have a better comparison of the results, in Table 2 we summarized the same calculations for our model trained with $\alpha = 1$ in the Appendix B. The model achieved an accuracy of **83.63%**, compared to the significantly lower accuracy of **40.83%** for the model trained with $\alpha = 1$. Accuracy is defined as $1-$ misclassification rate and represents the proportion of correctly predicted digits.

The strong difference in performance between the two models can be attributed to the impact of Laplace smoothing. The model with $\alpha = 2$ benefits from smoothing, preventing it from assigning zero probabilities to unseen pixel configurations, thereby improving its generalization ability. The model with $\alpha = 1$ overfits to the training data, leading to frequent misclassifications, particularly for digits with high variability, such as 5 and 8.

The performance metrics (precision, recall, and F1-score) further highlight the model's effectiveness. For the model with $\alpha = 2$, precision ranges from **0.75** (digit 9) to **0.91** (digit 0), indicating strong performance in distinguishing between classes, especially for distinct digits like 0 and 1. Recall scores range from **0.67** (digit 5) to **0.97** (digit 1), reflecting the model's ability to correctly identify the true instances of most digits. F1-scores show balanced performance, with values between **0.72** and **0.91**.

In contrast, the model with $\alpha = 1$ performs poorly across all metrics, particularly for complex digits like 5 and 8. Precision for digit 0 is as low as **0.17**, indicating frequent misclassifications, while digit 1, despite having high precision, suffers from low recall due to overfitting, reducing its overall F1-score.

3.2. Part b) Analysis of Misclassified Examples for $\alpha = 2$ and $\alpha = 1$

The errors produced by the model with $\alpha = 2$ show the strengths and weaknesses of Naive Bayes with Laplace smoothing. While the model achieved an accuracy of 83.63%, it still struggles with digits that share similar pixel configurations. In contrast, the model with $\alpha = 1$ exhibits significantly worse performance, with an accuracy of 40.83%, due to overfitting and the lack of smoothing. Below, we focus on the types of errors made by the model and the impact of α on these results. Even though, we computed all outputs both times, once for $\alpha = 2$ and once for $\alpha = 1$, for a detailed comparison with the output of our model trained with $\alpha = 1$ we refer to the Appendix B.2, in which we explained our evaluation for the model with $\alpha = 1$ in more detail.

3.2.1. Digits Grouped by Predicted Label: Model with $\alpha = 2$

As Figure 1 in the Appendix B.1 shows, the model trained with $\alpha = 2$ shows strong performance across most digits, but certain types of misclassifications occur more frequently.

- **Class 4**: Misclassified as 6, 8 and especially often 9. The model confuses rounded versions of 4 with 9 due to pixel overlap in critical regions. The independence assumption forces the classifier to rely solely on pixel intensities, causing errors when subtle handwriting differences activate unexpected pixels.

- **Class 5**: Often misclassified as 0 or 3, particularly when written with curved strokes. The model's inability to capture the structure of digits leads to confusion between 5 and the similarly shaped curves in the digits 0 and 3.

- **Class 8**: Commonly misclassified as 3 and 5 due to their similar loop structure. When the loop of a 8 is stretched to the sides, the model struggles to distinguish between the digits with partial loops.

3.2.2. Misclassified Digits: Model with $\alpha = 2$

When we focus exclusively on the misclassified digits in Figure 2 in Appendix B.1, the most notable errors involve digits that share structural similarities:

- **Class 2**: Often confused with 6, 7, and 8. The model mixes these digits when their top-right curves resemble each other, further highlighting the limitations of pixel-wise independence.

- **Class 9**: Misclassified as 0, 4 and 7, especially when the digit's lower end is not well-formed. The misclassification stems from the model's failure to account for overall shape.

These errors highlight that treating pixels as independent features, the model is sensitive to local variations but cannot integrate this information into an understanding of the digit's structure. Smoothing with $\alpha = 2$ improves the model's ability to handle unseen pixel configurations but does not eliminate this fundamental limitation. Especially the classes of 1, 3, 8 and 9 have the most errors in them as Figure 3 in Appendix B.1 shows.

3.2.3. Confusion Matrix: $\alpha = 2$ vs. $\alpha = 1$

From the confusion matrix shown in Figure 4 in Appendix B.1 of the model with $\alpha = 2$, we observe the following:

- **Highest Precision**: Digit 1 has the highest precision at 0.91, reflecting its simple structure and distinct pixel patterns, which reduce the likelihood of misclassification.
- **Lowest Precision**: Digit 9 has the lowest precision at 0.75, primarily due to its frequent confusion with digits 4 and 8, highlighting the challenges in distinguishing digits with shared pixel regions.
- **Highest Recall**: Digit 1 also achieves the highest recall at 0.97, confirming its easy recognizability.
- **Lowest Recall**: Digit 5 has the lowest recall at 0.67, as it is often misclassified as 0, 8, or 9, reflecting its structural ambiguity.

For $\alpha = 1$, the confusion matrix reveals significantly worse performance across all digits. Digit 0, in particular, is frequently predicted incorrectly, with many instances of 2, 3, and 8 misclassified as 0. This occurs because the lack of smoothing prevents the model from generalizing to unseen pixel configurations, leading to overfitting and a higher number of misclassifications across all classes.

3.2.4. Theoretical Explanation of Errors

The observed misclassifications comes from fundamental Naive Bayes assumptions:

- **Independence Assumption**: The Naive Bayes classifier assumes each pixel is independent. This is unrealistic for images, where spatial relationships define digit structure, leading to struggles with overlapping or ambiguous configurations, particularly for digits like 5, 8, and 9.
- **Effect of Smoothing**: With $\alpha = 2$, smoothing prevents zero probabilities for unseen configurations, enhancing the model's ability to handle handwriting variations. However, the independence assumption still restricts its capability to distinguish similar digits, as smoothing does not fully account for overall digit structure.

In the model with $\alpha = 1$, insufficient smoothing results in overfitting, where the model relies too heavily on training pixel configurations. This leads to poor generalization for complex or curved digits.

In conclusion, the performance of the Naive Bayes classifier is heavily influenced by the value of α. The model with $\alpha = 2$ balances preventing zero probabilities for unseen configurations and maintaining generalization. However, the independence assumption limits performance, particularly for digits with shared visual components, leading to common misclassifications. Conversely, the model with $\alpha = 1$ suffers significant overfitting, underscoring the importance of smoothing for effective generalization.

4. Task 4 - Model Selection: Find the best value for α

In this task, we use 5-fold cross-validation to identify the optimal value for the smoothing parameter α in the Naive Bayes model. The tested α values were $\alpha = [0.5, 1, 1.3, 1.5, 1.75, 2, 2.5, 3, 5, 10]$, and the cross-validated accuracy for each value is summarized in C.

The model achieved its highest accuracy of **83.40%** with $\alpha = 1.3$, indicating that moderate smoothing is most effective. As α increases beyond 1.75, the accuracy begins to decline, suggesting that excessive smoothing leads to underfitting, where the model loses important data patterns. In contrast, the lowest accuracy of **30.34%** occurred with $\alpha = 1$, highlighting the dangers of under-smoothing and overfitting.

At $\alpha = 1.3$, the model finds an optimal balance: it avoids assigning zero probabilities to unseen pixel combinations while maintaining enough specificity to classify the digits accurately. This result aligns with the theory behind Laplace smoothing, which mitigates overfitting by handling unseen data while still allowing the model to learn effectively.

In conclusion, $\alpha = 1.3$ proves to be the best tested choice for this Naive Bayes model on the MNIST dataset, preventing both overfitting and underfitting. Models with $\alpha > 2$ show diminishing returns, with the steep decline at $\alpha = 10$ indicating that too much smoothing reduces the model's predictive ability.

5. Task 5 b) - Generating Digits

In this section, we analyze the generated digits for the model with $\alpha = 2$, for the model with $\alpha = 1$ and for the model with the in Section 4 as an optimal α-value tested $\alpha = 1.3$. The different values of the hyperparameter α control the level of Laplace smoothing in the Naive Bayes model. Using the function *nb_generate*, we generate digits for a given class label and visualize how it influences the diversity and quality of the generated digits, as well as the sharpness and clarity of the features.

5.1. Generated Digits for $\alpha = 1$

For $\alpha = 1$, which corresponds to no Laplace smoothing, the generated digits in Figure 16, 17 and 18 in Appendix D.2 show significant noise and distortion. The lack of sufficient smoothing means that the model overfits to the training data, leading to pixel configurations that reflect over-reliance on specific patterns seen during training rather than generalized representations of each digit class.

- Digit 8: In particular, digit 8 appears poorly formed, with broken loops and erratic pixel values. This suggests that the model has failed to generalize a consistent shape for the digit, instead relying on specific instances from the training data that do not fully capture the general structure of an 8.
- Digit 5: Similar issues are seen with digit 5, where the model fails to create a clear distinction between the top and bottom halves of the digit. The overfitting is most evident in how the bottom part blends with the background.

5.2. Generated Digits for $\alpha = 2$

Increasing the smoothing parameter to $\alpha = 2$ improves the quality of generated digits (see Figures 13, 14, and 15 in Appendix D.1). The model can generalize better, leading to more recognizable and structured digits.

- **Digit 3**: The digit 3 is well-formed, with a distinct curvature and visible features, suggesting that the model captures the essential characteristics.
- **Digit 9**: Digit 9 is mostly accurate, though the loop is slightly open, indicating residual uncertainty. This represents a significant improvement over the noisy outputs for $\alpha = 1$.

While the digits are generally clear at $\alpha = 2$, slight blurring remains, reflecting a balance between generalization and specificity.

5.3. Generated Digits for $\alpha = 1.3$ (Optimal Value)

With $\alpha = 1.3$, the model achieves an optimal balance, as shown in Figures 19, 20, and 21 in Appendix D.3. The digits generated at this setting are sharper and more accurate than those with $\alpha = 1$, while being less smoothed than those with $\alpha = 2$.

- **Digit 1**: This digit appears crisp and well-defined, showcasing the model's ability to generalize without losing precision.
- **Digit 8**: Compared to the noisy representation at $\alpha = 1$, the digit 8 generated at $\alpha = 1.3$ displays more complete loops and a clearer shape.

The analysis of the generated digits reveals the critical role that the hyperparameter α plays in controlling the balance between overfitting and underfitting in the Naive Bayes model. Lower values of α (such as 1) result in overfitting, producing noisy and distorted digits, while higher values (such as 2) lead to smoother but slightly less sharp digits. The optimal value of $\alpha = 1.3$ provides the best overall performance, generating digits that are both sharp and accurate, with minimal noise. This highlights the importance of carefully tuning the smoothing parameter to achieve the best results in generative tasks.

6. Task 6 - Missing Data

The definition of all objects used can be seen in the appendix A. The calculation in a) and b) are just Proposition A.1 where for all $i \in \{1, ..., \dim(\Theta)\}$ we denote $X_i := \gamma_i(X)$. For $k_1, ..., k_D \in \{0, ..., K-1\}$ and $\vartheta \in \Theta$ we have
a)

$$\begin{aligned}\mathbb{P}(\theta = \vartheta \mid X_1 = k_1, ..., X_D = k_D) &= \frac{\prod_{i=1}^{D} \mathbb{P}(X_i = k_i \mid \theta = \vartheta) \cdot \mathbb{P}(\theta = \vartheta)}{\mathbb{P}(X_1 = k_1, ..., X_D = k_D)} \\ &\propto \prod_{i=1}^{D} \mathbb{P}(X_i = k_i \mid \theta = \vartheta) \cdot \mathbb{P}(\theta = \vartheta),\end{aligned}$$

Interpretation: High mass, where the probability that a pixel has value k_i given ϑ is high for every $i = 1, ..., D$ and the prior belief is also high. If one conditional probability is very small, then the entire probability is small, due to the product and vice versa.
b) The calculation is analougously only that for $D' < D$:

$$\begin{aligned}\mathbb{P}(\theta = \vartheta \mid X_1 = k_1, ..., X_{D'} = k_{D'}) &= \frac{\prod_{i=1}^{D'} \mathbb{P}(X_i = k_i \mid \theta = \vartheta) \cdot \mathbb{P}(\theta = \vartheta)}{\mathbb{P}(X_1 = k_1, ..., X_{D'} = k_{D'})} \\ &\propto \prod_{i=1}^{D'} \mathbb{P}(X_i = k_i \mid \theta = \vartheta) \cdot \mathbb{P}(\theta = \vartheta),\end{aligned}$$

Interpretation: Same interpretation, only that we only consider the product until D'.
c)

$$\begin{aligned}&\mathbb{P}(X_{D'+1} = k_{D'+1}, ..., X_D = k_D \mid X_1 = k_1, ..., X_{D'} = k_{D'}) \\ &= \sum_{\vartheta \in \Theta} \frac{\mathbb{P}(X_1 = k_1, ..., X_D = k_D, \theta = \vartheta)}{\mathbb{P}(X_1 = x_1, ..., X_{D'} = k_{D'})} \\ &= \sum_{\vartheta \in \Theta} \frac{\mathbb{P}(\theta = \vartheta \mid X_1 = k_1, ..., X_D = k_D)}{\mathbb{P}(X_1 = x_1, ..., X_{D'} = k_{D'})} \cdot \mathbb{P}(X_1 = k_1, ..., X_D = k_D) \\ &= \sum_{\vartheta \in \Theta} \frac{\prod_{i=1}^{D} \mathbb{P}(X_i = k_i \mid \theta = \vartheta) \mathbb{P}(\theta = \vartheta)}{\mathbb{P}(X_1 = k_1, ..., X_D = k_D)} \cdot \frac{\mathbb{P}(X_1 = k_1, ..., X_D = k_D)}{\mathbb{P}(X_1 = k_1, ..., X_{D'} = k_{D'})} \\ &= \sum_{\vartheta \in \Theta} \frac{\prod_{i=1}^{D} \mathbb{P}(X_i = k_i \mid \theta = \vartheta) \mathbb{P}(\theta = \vartheta)}{\mathbb{P}(X_1 = k_1, ..., X_{D'} = k_{D'})}\end{aligned}$$

Interpretation: The distribution of pixels $D' + 1$ until D given pixels 1 to D' has a lot of mass where the probability $\mathbb{P}(X_i = k_i \mid \theta = \vartheta)$ is high for all $i \in \{D' + 1, \dots, D\}$. This means that when the conditional probability of observing specific values for the remaining pixels is high given a particular class ϑ whose occurence was already partially up to D respected, the overall likelihood of those pixel values occurring in that class dominates the distribution. As a result, the observed subset of pixels (1 to D') provides information about the remaining pixel values ($D' + 1$ to D) when not all features have been observed, and one wants to make predictions or perform inference based on the observed subsets of the data.

7. Conclusion

This report analyzed the Naive Bayes classifier on the MNIST dataset, focusing on the impact of the Laplace smoothing parameter α. Cross-validation identified $\alpha = 1.3$ as the optimal value, balancing generalization and precision to achieve high classification accuracy and clearer generated digits.

We also observed the limitations of the Naive Bayes assumption of feature independence, particularly in handling spatial relationships between pixels, which led to errors for visually similar digits. Furthermore, experiments with digit generation showed that insufficient smoothing causes overfitting, while excessive smoothing leads to overly generalized results.

In conclusion, while simple, the Naive Bayes model proves effective for both classification and data generation when key hyperparameters, such as α, are properly tuned.

References

A. Statement and derivation of Solution of Task 1 and 2

A.1. Task 1

Definition A.1. *For all $i \in \{1, ..., dim(\mathfrak{X})\}$ consider the mapping $\gamma_i : \mathfrak{X} \to \{0, ..., K - 1\}, x \mapsto x_i$ which is the projection of a picture to an individual pixel.*

We are interested for the first task of the distribution of the prior, i.e.

$$\pi(A) := \sum_{\vartheta \in \Theta} p_\vartheta \mathbf{1}_{\vartheta \in A}, \quad \text{where } \sum_{\vartheta \in \Theta} p_\vartheta = 1,$$

because the underlying space is discrete. In the task we set the prior distribution to the symmetric Dirichlet distribution, i.e. we set for $\alpha \in \mathbb{R}_+^{dim(\Theta)}$

$$\forall \vartheta \in \Theta : p_\vartheta := \frac{1}{B(\alpha)} \prod_{\vartheta \in \Theta} \vartheta^{\alpha_\vartheta - 1} \quad \text{and } B(\alpha) := \sum_{\vartheta \in \Theta} \prod_{\vartheta \in \Theta} \vartheta^{\alpha_\vartheta - 1}.$$

In our case $dim(\Theta) = 1$. Now for the function nb-train all we do is take a sample from π, which in the case of $\alpha = 1$ is just the uniform distribution.
Finally, we were supposed to sample from the distribution of a pixel given a class/parameter ϑ, which is the posterior distribution, i.e. for all $A \in \mathcal{P}(\{0, ..., K - 1\})$

$$\begin{aligned} Q^{\gamma(X)|\theta=\vartheta}(A) = \mathbb{P}(\gamma(X) \in A \mid \theta = \vartheta) &\overset{\text{Bayes}}{=} \frac{\mathbb{P}(\gamma(X) \in A, \theta = \vartheta)}{\mathbb{P}(\theta = \vartheta)} \\ &= \frac{\mathbb{P}(\theta = \vartheta \mid \gamma(X) \in A) \cdot \mathbb{P}(\gamma(X) \in A)}{\mathbb{P}(\theta = \vartheta)}. \end{aligned} \tag{1}$$

A.2. Task 2

Remark A.1. *The goal of the Naive Bias classifier is to solve*

$$\hat{\theta}_{MAP}(X_1, ..., X_n) \in argmax_{\theta \in \Theta} \mathbb{P}(\theta = \vartheta \mid \gamma(X_1) = k_1, ..., \gamma(X_n) = k_n).$$

There are two approaches. Either sample from it for every $\vartheta \in \Theta$, by using Proposition A.1, because these Probabilites can be sampled. The second option is to calculate the argmax from the probability weights (all spaces are discrete). In Exercise Nr1 we determined $\hat{\theta}_{MAP}$ by calculating the argmax, i.e. maximizing the probability weight function.

Remark A.2. *Let $\theta \sim Dir(\alpha)$ denote the Dirichlet distribution and $X \sim Mu(\vartheta := (\vartheta_1, ..., \vartheta_K))$, then as in the lecture notes we can can just calculate using basic methods from analysis the maximum of the product density weight functions of these two distributions, which is*

$$\frac{1}{n + \alpha_0 - K} \cdot \begin{pmatrix} n_1 + \alpha_1 - 1 \\ \cdots \\ n_K + \alpha_K - 1 \end{pmatrix} \in argmax_{\theta \in \Theta} \mathbb{P}(\theta = \vartheta \mid \gamma(X_1) = k_1, ..., \gamma(X_n) = k_n),$$

where $n = \sum_{i=1}^{K} n_i$ and $\alpha_0 = \sum_{i=1}^{K} \alpha_i$.

Remark A.3. *The second method is taking the probability $\mathbb{P}(\theta = \vartheta)$ (which is known) and $\mathbb{P}(\gamma(X_i) = k_i \mid \theta = \vartheta)$ can be rewritten with (1) as*

$$\mathbb{P}(\gamma(X_i) = k_i \mid \theta = \vartheta) = \frac{\mathbb{P}(\theta = \vartheta \mid \gamma(X) \in A) \cdot \mathbb{P}(\gamma(X) \in A)}{\mathbb{P}(\theta = \vartheta)}.$$

All these probabilities can be approximated by the discrete empirical distributions, i.e. for samples $\theta_1, ..., \theta_n$ and $X_1, ..., X_n$ we define

$$F_n(t) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\{\gamma(X_i) = k\}}, \quad k \in \{0, ...K - 1\}$$

as the empirical distribution and the conditional empirical distribution

$$F_n(t) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\{\theta = \vartheta \mid \gamma(X_i) = k\}}, \quad k \in \{0, ...K - 1\}.$$

A.3. Task 6

Proposition A.1. *Consider a Bayesian Model $(\mathfrak{X}, \Theta, Q)$ as in Definition 1.2 and for all $i \in \{0, ..., n\}$ the mapping γ_i from Definition A.1, where $n = dim(\mathfrak{X})$, then for $\vartheta \in \Theta$ and $k_1, ..., k_n \in \{0, ..., K - 1\}$:*

$$\mathbb{P}(\theta = \vartheta \mid \gamma_1(X) = k_1, ..., \gamma_n(X) = k_N) \propto \prod_{i=1}^{n} \mathbb{P}(\gamma_i(X) = k_i \mid \theta = \vartheta) \cdot \mathbb{P}(\theta = \vartheta)$$

although we look at the proportionality wrt. θ.

Proof. We can rewrite it as

$$
\begin{aligned}
&\mathbb{P}(\theta = \vartheta \mid \gamma_1(X) = k_1, ..., \gamma_n(X) = k_n) \\
&= \frac{\mathbb{P}(\theta = \vartheta, \gamma_1(X) = k_1, ..., \gamma_n(X) = k_n)}{\mathbb{P}(\gamma(X_1) = k_1, ..., \gamma_n(X) = k_n)} \\
&= \frac{\mathbb{P}(\gamma_1(X) = k_1 \mid \theta = \vartheta, \gamma_2(X) = k_2, ..., \gamma_n(X) = k_n) \cdot \mathbb{P}(\theta = \vartheta, \gamma_2(X) = k_2, ..., \gamma_n(X) = k_n)}{\mathbb{P}(\gamma_1(X) = k_1, ..., \gamma_n(X) = k_n)} \\
&= \frac{\prod_{i=1}^{n-1} \mathbb{P}(\gamma_i(X) = k_i \mid \gamma_{i+1}(X) = k_{i+1}, ..., \gamma_n(X) = k_n, \theta = \vartheta) \cdot \mathbb{P}(\gamma_n(X) \mid \theta = \vartheta)\mathbb{P}(\theta = \vartheta)}{\mathbb{P}(\gamma_1(X) = k_1, ..., \gamma_n(X) = k_n)}
\end{aligned}
$$

With the naive Bias assumption the above is equal to

$$
\begin{aligned}
\mathbb{P}(\theta = \vartheta \mid \gamma_1(X) = k_1, ..., \gamma_n(X) = k_n) &= \frac{\prod_{i=1}^{n} \mathbb{P}(\gamma_i(X) = k_i \mid \theta = \vartheta) \cdot \mathbb{P}(\theta = \vartheta)}{\mathbb{P}(\gamma_1(X) = k_1, ..., \gamma_n(X) = k_n)} \\
&\propto \prod_{i=1}^{n} \mathbb{P}(\gamma_i(X) = k_i \mid \theta = \vartheta) \cdot \mathbb{P}(\theta = \vartheta),
\end{aligned}
$$

although we look at the proportionality wrt. θ. □

B. Plots of digits for predicted label class

B.1. Model with $\alpha = 2$

In the following, the plots and outputs of calculating the evaluation metrics produced by our code are shown for the *model_nb2* trained with add-one-smoothing $\alpha = 2$.

Class	Precision	Recall	F1-Score	Support
0	0.91	0.89	0.90	980
1	0.86	0.97	0.91	1135
2	0.89	0.79	0.84	1032
3	0.77	0.83	0.80	1010
4	0.82	0.82	0.82	982
5	0.78	0.67	0.72	892
6	0.88	0.89	0.89	958
7	0.91	0.84	0.87	1028
8	0.79	0.78	0.79	974
9	0.75	0.85	0.80	1009
Accuracy	0.84			10000
Macro avg	0.84	0.83	0.83	10000
Weighted avg	0.84	0.84	0.84	10000

Table 1: Values of Accuracy, Precision, Recall and F1-Score for model trained with $\alpha = 2$

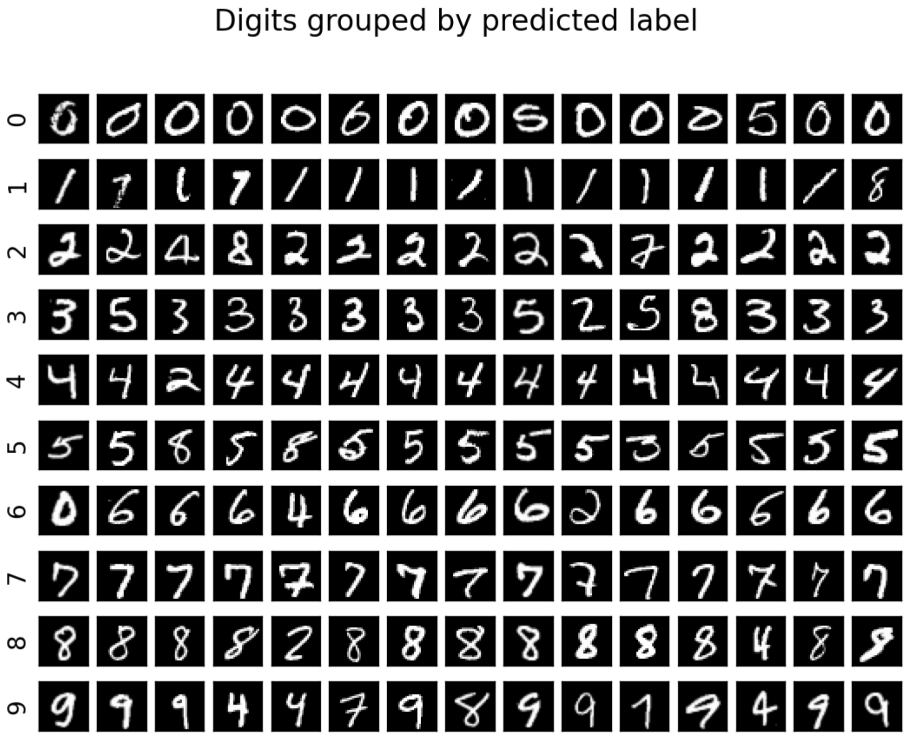


Figure 1: Digits grouped by predicted label. Also misclassified digits are shown among the predictions for the model with $\alpha = 2$.

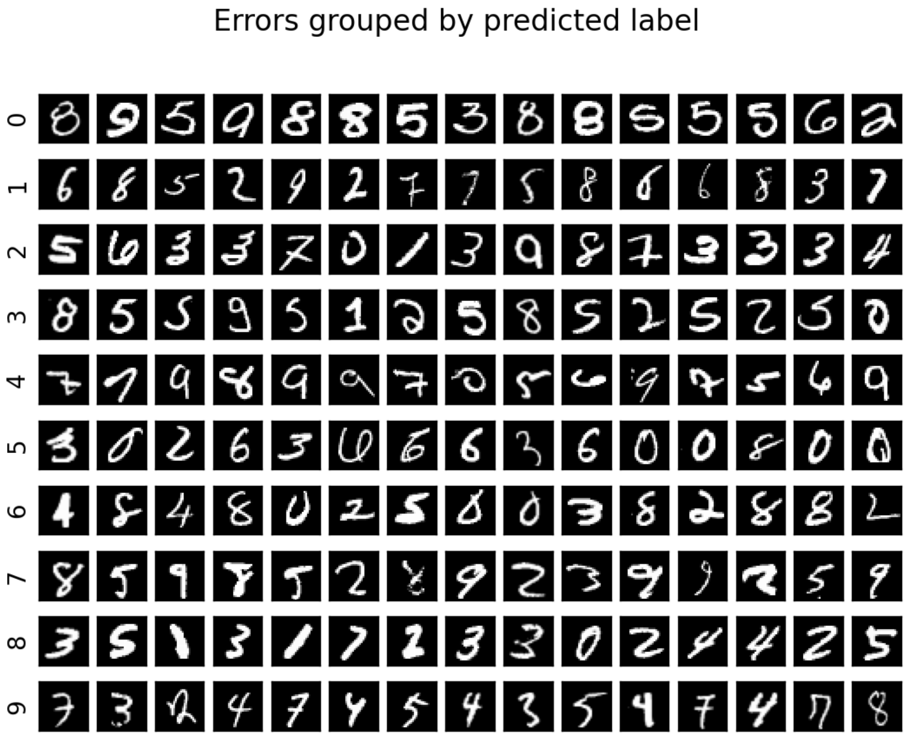


Figure 2: Errors grouped by predicted label. Only misclassified predictions are shown for the model with $\alpha = 2$.

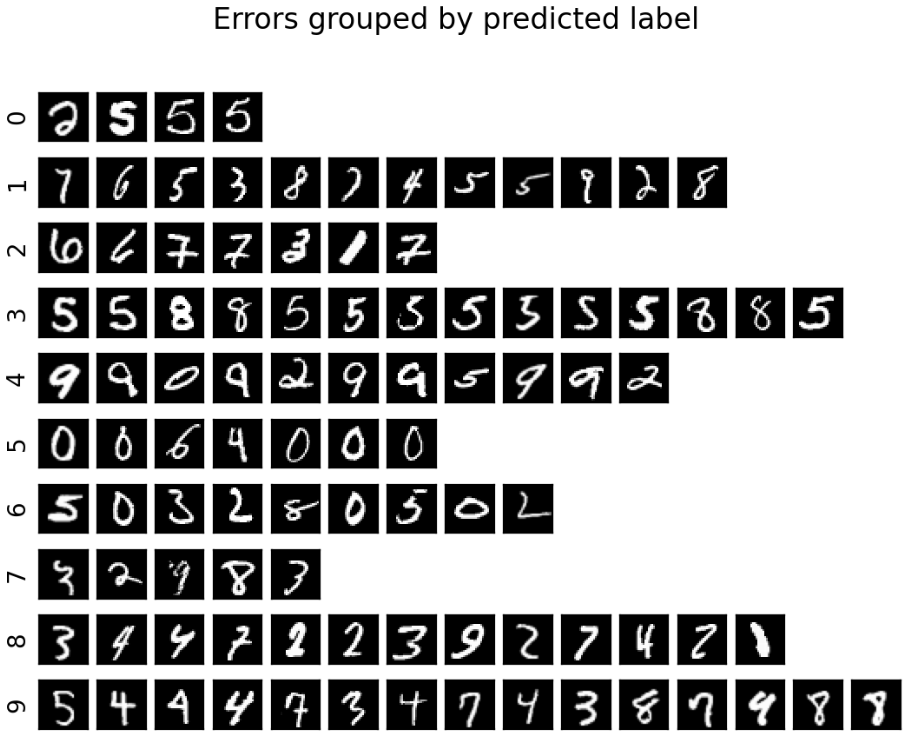
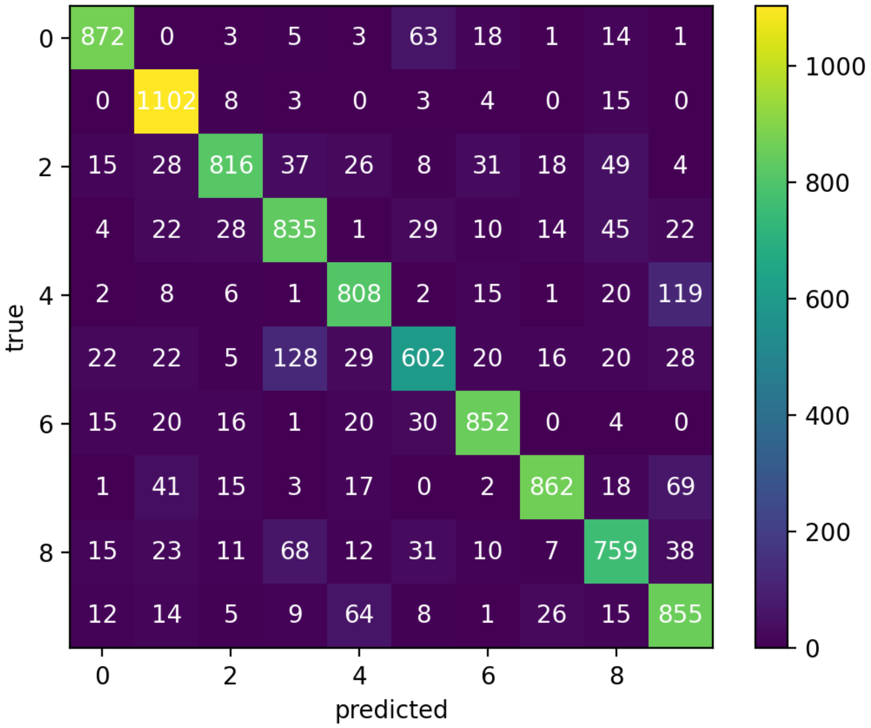


Figure 3: Errors grouped by predicted label are shown in the corresponding proportion of each class they have been misclassified on for the model with $\alpha = 2$.



	0		2		4		6		8	
0	872	0	3	5	3	63	18	1	14	1
	0	1102	8	3	0	3	4	0	15	0
2	15	28	816	37	26	8	31	18	49	4
	4	22	28	835	1	29	10	14	45	22
4	2	8	6	1	808	2	15	1	20	119
	22	22	5	128	29	602	20	16	20	28
6	15	20	16	1	20	30	852	0	4	0
	1	41	15	3	17	0	2	862	18	69
8	15	23	11	68	12	31	10	7	759	38
	12	14	5	9	64	8	1	26	15	855

Figure 4: Confusion matrix for the model with $\alpha = 2$

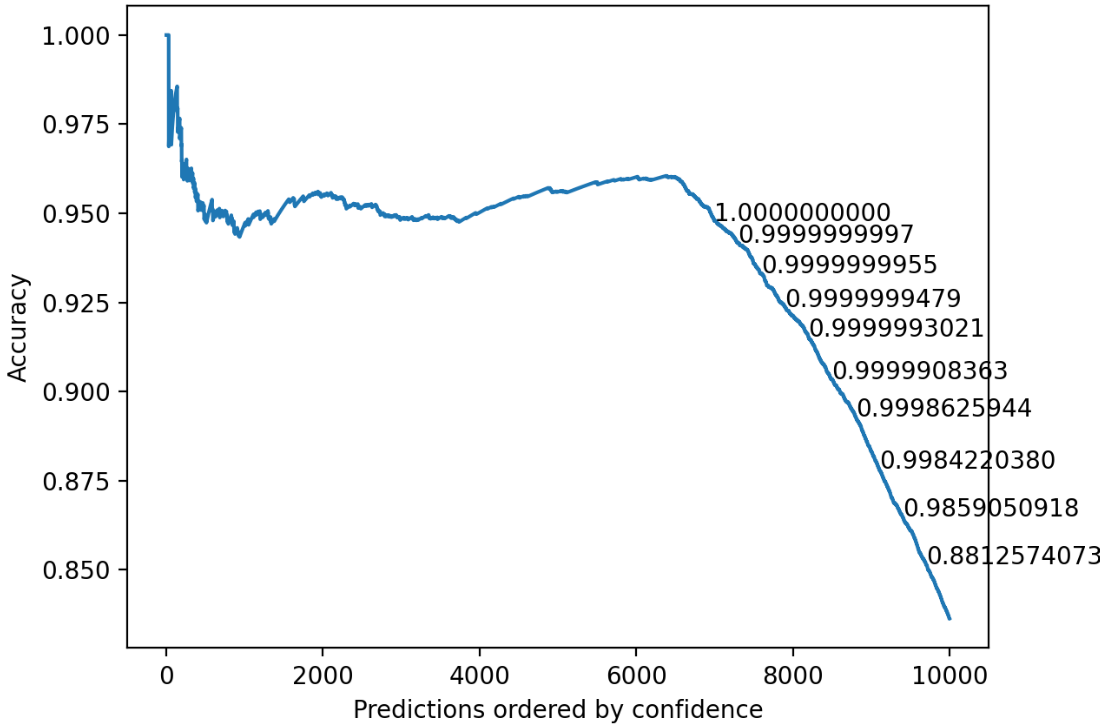


Figure 5: Cumulative accuracy for predictions ordered by confidence for the model with $\alpha = 2$

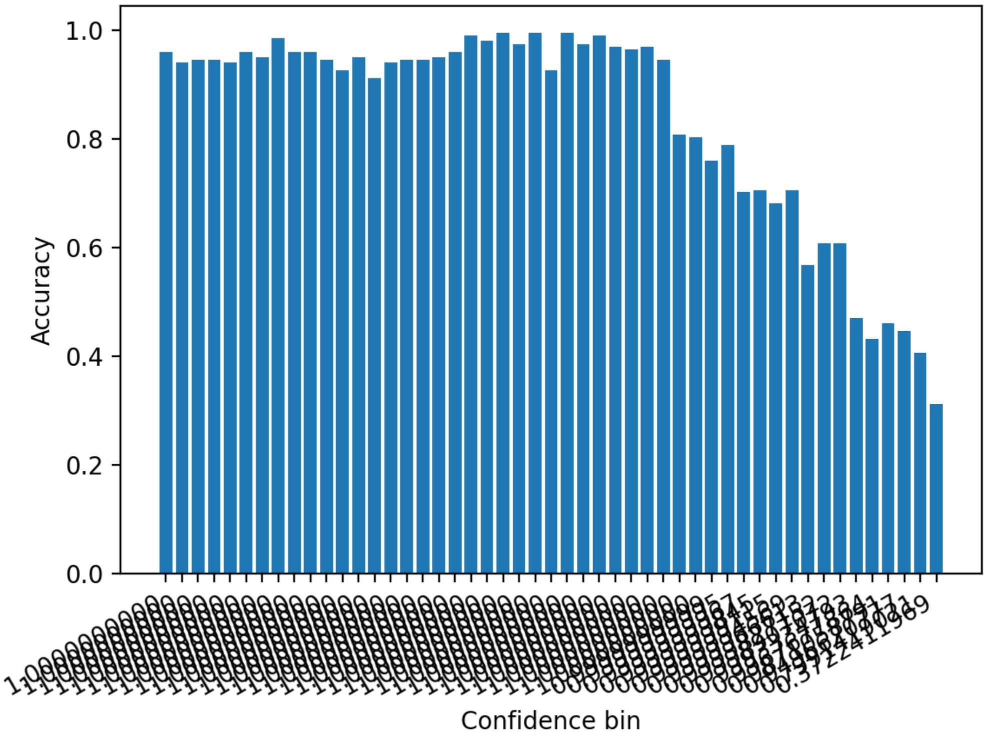


Figure 6: Accuracy for predictions grouped by confidence for the model with $\alpha = 2$

B.2. Model with $\alpha = 1$

B.2.1. Comparison with Model Trained with $\alpha = 1$

The performance of the model trained with $\alpha = 1$ is significantly worse as one can see in Figures 7, 8, 9 in Appendix B.2 with an overall accuracy of 40.83% as presented in Table 2 in Appendix B.2. This drop is largely due to the absence of effective smoothing, which causes the model to assign zero probabilities to unseen pixel configurations. Without sufficient smoothing, the model cannot generalize to new or slightly altered pixel patterns in the test set.

- **Class 0**: In the model with $\alpha = 1$, almost all predictions for digit 0 are incorrect, with many instances of digits 2, 3, and 8 being misclassified as 0. This occurs because, without smoothing, the model cannot handle the variations in handwriting for these digits and defaults to incorrect predictions.

- **Class 5**: Precision for digit 5 is extremely low (0.37), and recall is also poor (0.23). The model frequently misclassifies 5 as 0, 8, or 9, similar to the model with $\alpha = 2$, but the errors are far more frequent due to the lack of smoothing.

- **General behavior**: Across most classes, precision and recall are significantly lower for the model with $\alpha = 1$. The model cannot handle variations in pixel patterns as effectively, leading to widespread misclassifications, particularly for digits with curved or complex structures like 5, 8, and 9.

The confusion matrix for $\alpha = 1$ shows that the model is heavily biased towards certain predictions, often defaulting to incorrect labels when it encounters unfamiliar pixel configurations. This demonstrates the importance of using adequate smoothing to prevent overfitting to the training data.

B.2.2. Analysis of Confidence vs. Accuracy: $\alpha = 2$ vs $\alpha = 1$

The confidence-accuracy plots in Figures 5, 6 in Appendix B.1 and Figures 11 and 12 in Appendix B.2 illustrate how well-calibrated the Naive Bayes models are based on predicted confidence.

For the model with $\alpha = 2$, there is a strong positive correlation between confidence and accuracy. Predictions made with high confidence (close to 1) typically correspond to correct classifications, achieving nearly 100% accuracy in the top confidence bins. This indicates that the model is well-calibrated for confident predictions, and while accuracy decreases with lower confidence, the drop is smooth, suggesting reliability even when less confident. This pattern aligns with the effects of Laplace smoothing. With $\alpha = 2$, the model avoids assigning zero probabilities to unseen pixel configurations, which helps prevent overfitting and allows for better generalization. As a result, the model remains accurate at moderate confidence levels, managing uncertainty effectively. However, at lower confidence levels, accuracy drops significantly due to the model encountering test instances with pixel patterns less aligned with training data. While smoothing mitigates catastrophic failures, the independence assumption limits the model's ability to recognize spatial relationships between pixels, leading to misclassifications for ambiguous patterns.

In contrast, the confidence-accuracy plot for $\alpha = 1$ shows a less stable relationship. Although high-confidence predictions still yield reasonable accuracy, the decline in accuracy with decreasing confidence is much steeper, indicating that the model frequently assigns high confidence to incorrect predictions. This instability results from inadequate smoothing. The model with $\alpha = 1$ tends to assign zero probabilities to unseen pixel configurations, causing overfitting. Consequently, when faced with slightly different patterns, the model not only lowers its confidence but also experiences a sharp drop in accuracy, reflecting its inability to generalize beyond the training data. Additionally, for $\alpha = 1$, certain confidence bins exhibit much lower accuracy than expected, suggesting the model is overly confident in incorrect predictions, a common issue when overfitting occurs.

In the following, the plots and outputs of calculating the evaluation metrics produced by our code are shown for the *model_nb* trained without add-one-smoothing, so the choice of $\alpha = 1$.

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0	0.17	0.93	0.28	980
1	1.00	0.55	0.71	1135
2	0.64	0.24	0.35	1032
3	0.65	0.35	0.45	1010
4	0.77	0.25	0.38	982
5	0.37	0.23	0.28	892
6	0.93	0.34	0.50	958
7	0.92	0.34	0.50	1028
8	0.47	0.40	0.43	974
9	0.72	0.42	0.53	1009
Accuracy	0.41			10000
Macro avg	0.66	0.41	0.44	10000
Weighted avg	0.67	0.41	0.45	10000

Table 2: Values for Accuracy, Precision, Recall and F1-Score for model trained with $\alpha = 2$.

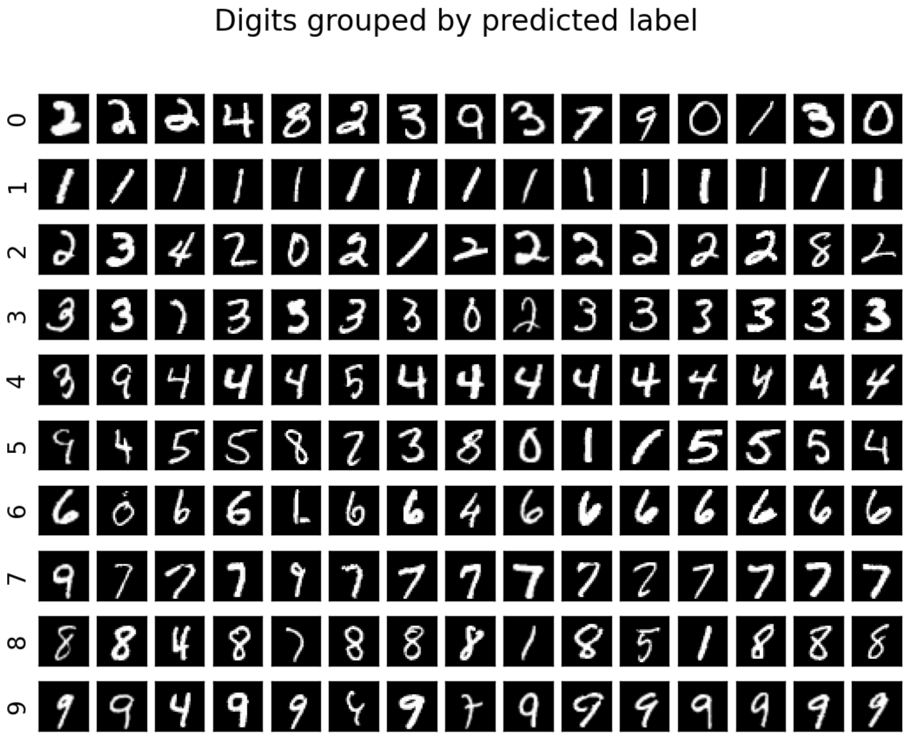


Figure 7: Digits grouped by predicted label. Also misclassified digits are shown among the predictions for the model with $\alpha = 1$.

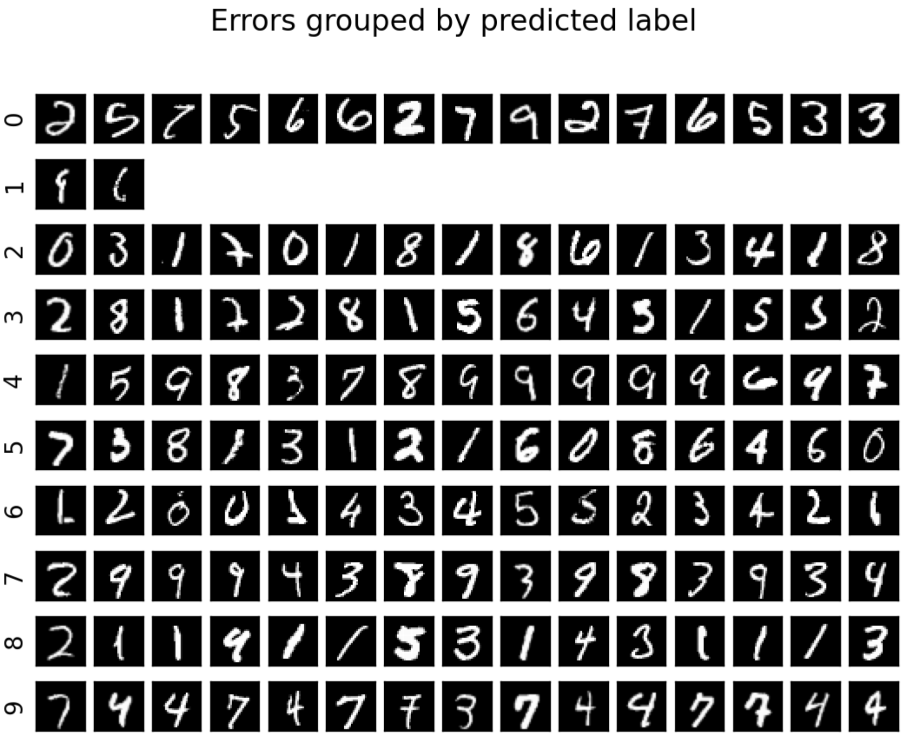


Figure 8: Errors grouped by predicted label. Only misclassified predictions are shown for the model with $\alpha = 1$.

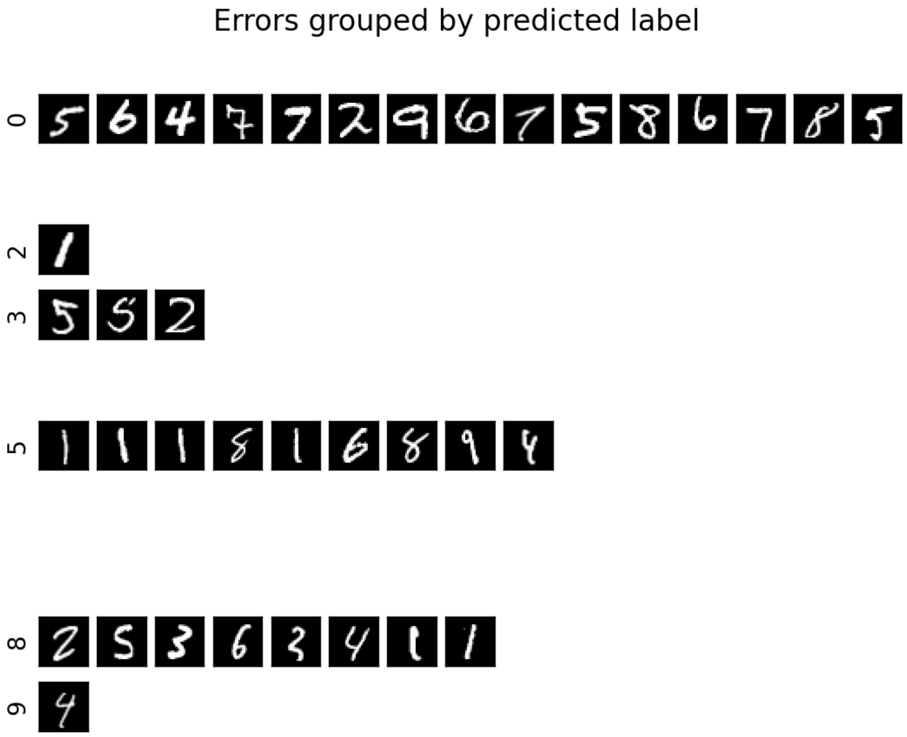
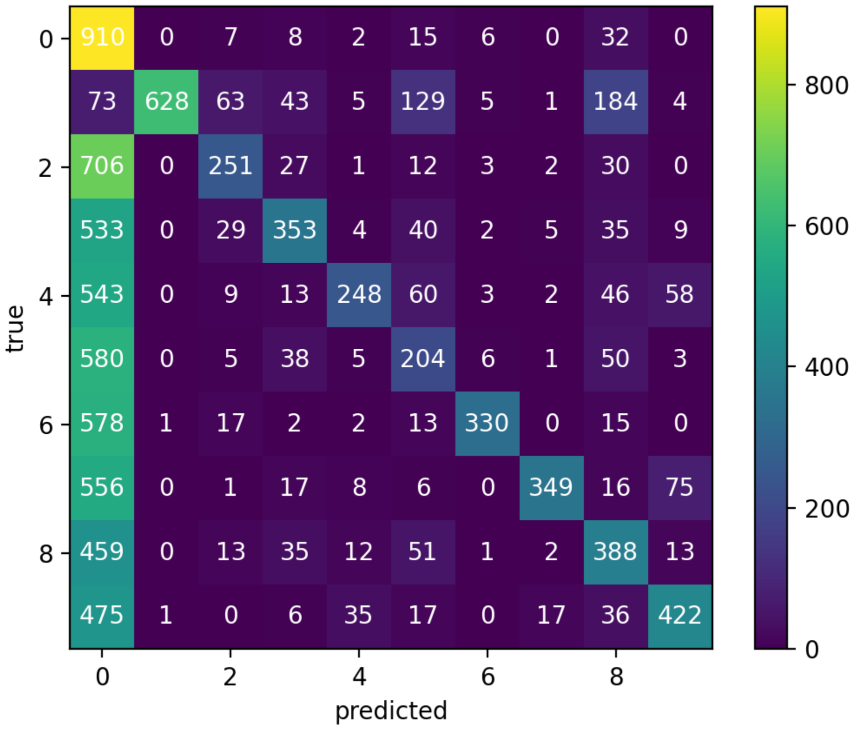


Figure 9: Errors grouped by predicted label are shown in the corresponding proportion of each class they have been misclassified on for the model with $\alpha = 1$.



	0		2		4		6		8	
0	910	0	7	8	2	15	6	0	32	0
	73	628	63	43	5	129	5	1	184	4
2	706	0	251	27	1	12	3	2	30	0
	533	0	29	353	4	40	2	5	35	9
4	543	0	9	13	248	60	3	2	46	58
	580	0	5	38	5	204	6	1	50	3
6	578	1	17	2	2	13	330	0	15	0
	556	0	1	17	8	6	0	349	16	75
8	459	0	13	35	12	51	1	2	388	13
	475	1	0	6	35	17	0	17	36	422

Figure 10: Confusion matrix for the model with $\alpha = 1$

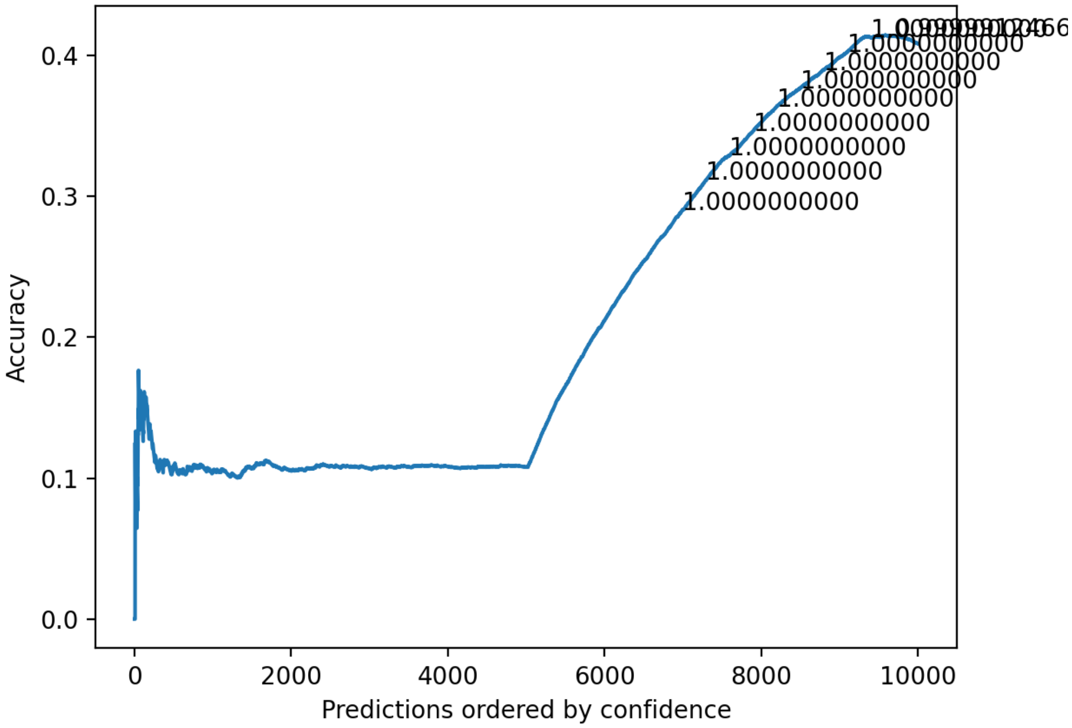


Figure 11: Cumulative accuracy for predictions ordered by confidence for the model with $\alpha = 1$

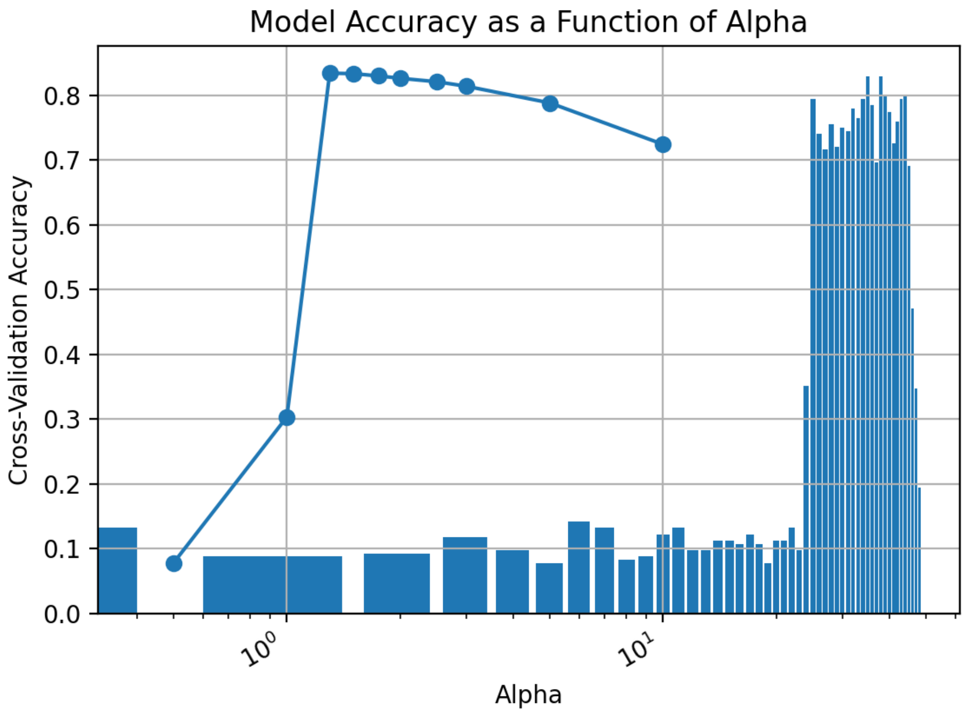


Figure 12: Accuracy for predictions grouped by confidence for the model with $\alpha = 1$

C. Model Selection: Accuracy for different α-values

We observed the following accuracies for each α:

- $\alpha = 0.5$: Average Accuracy = **0.0778** (to be computed if available)
- $\alpha = 1$: Average Accuracy = **0.3034**
- $\alpha = 1.3$: Average Accuracy = **0.8340**
- $\alpha = 1.5$: Average Accuracy = **0.8332**
- $\alpha = 1.75$: Average Accuracy = **0.8297**
- $\alpha = 2$: Average Accuracy = **0.8261**
- $\alpha = 2.5$: Average Accuracy = **0.8207**
- $\alpha = 3$: Average Accuracy = **0.8138**
- $\alpha = 5$: Average Accuracy = **0.7881**

- $\alpha = 10$: Average Accuracy = **0.7241**

D. Generating data

D.1. Model with $\alpha = 2$

In the following subsection, the model *model_nb2* with $\alpha = 2$ is analyzed.

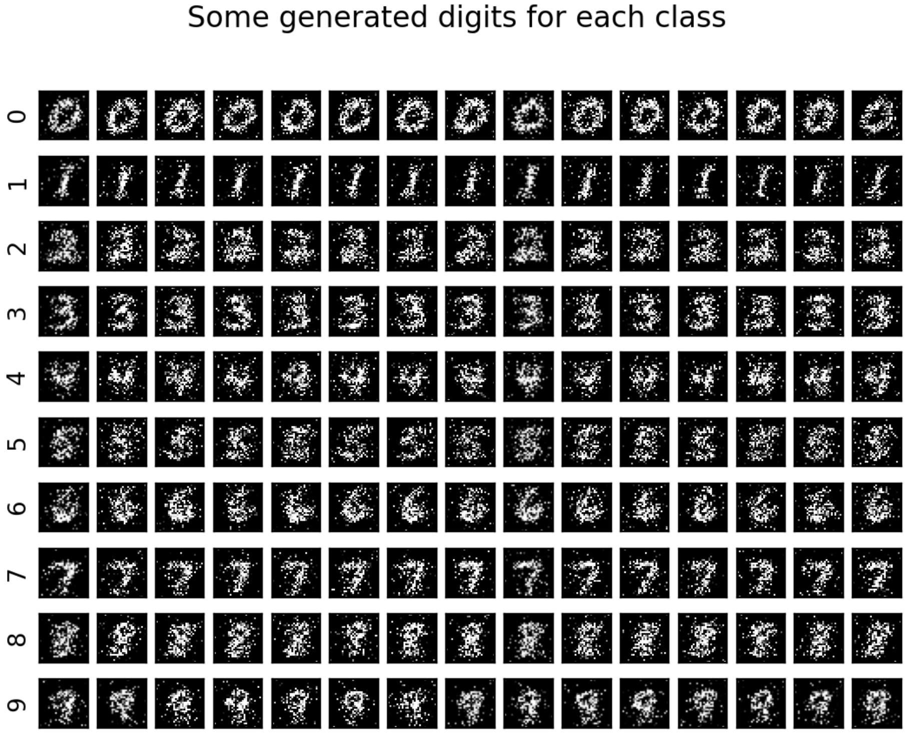


Figure 13: Generated digits for $\alpha = 2$. The digits are mostly well-formed and recognizable, with minor distortions in some classes.

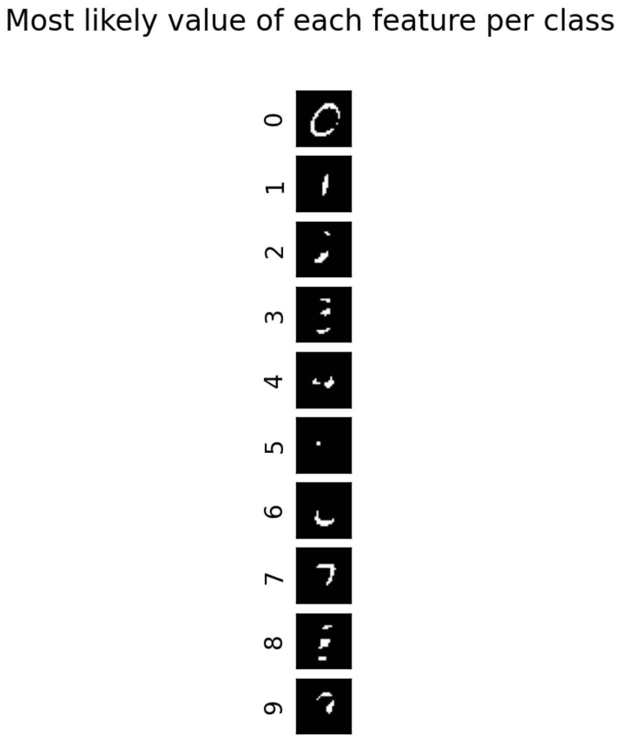


Figure 14: Most likely value for each feature per class for model with $\alpha = 2$.

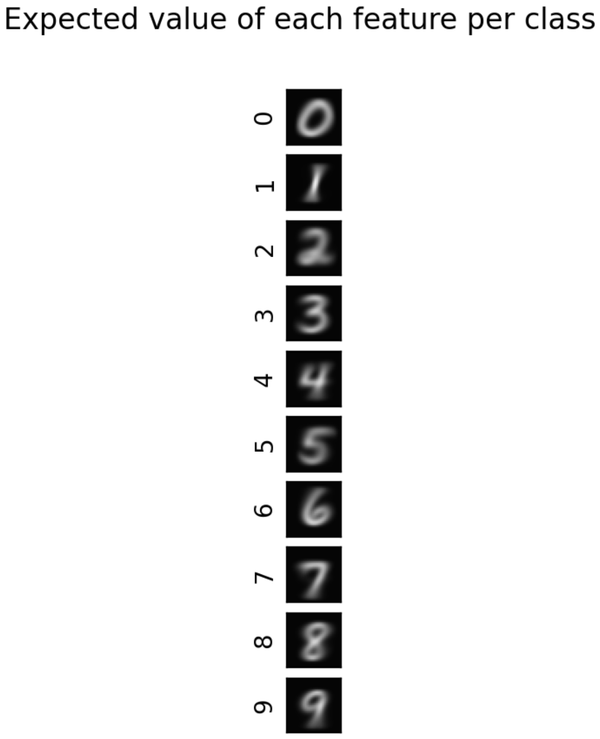


Figure 15: Expected value for each feature per class for model with $\alpha = 2$.

D.2. Model with $\alpha = 1$

In the following subsection, the model *model_nb* with $\alpha = 1$ is analyzed.

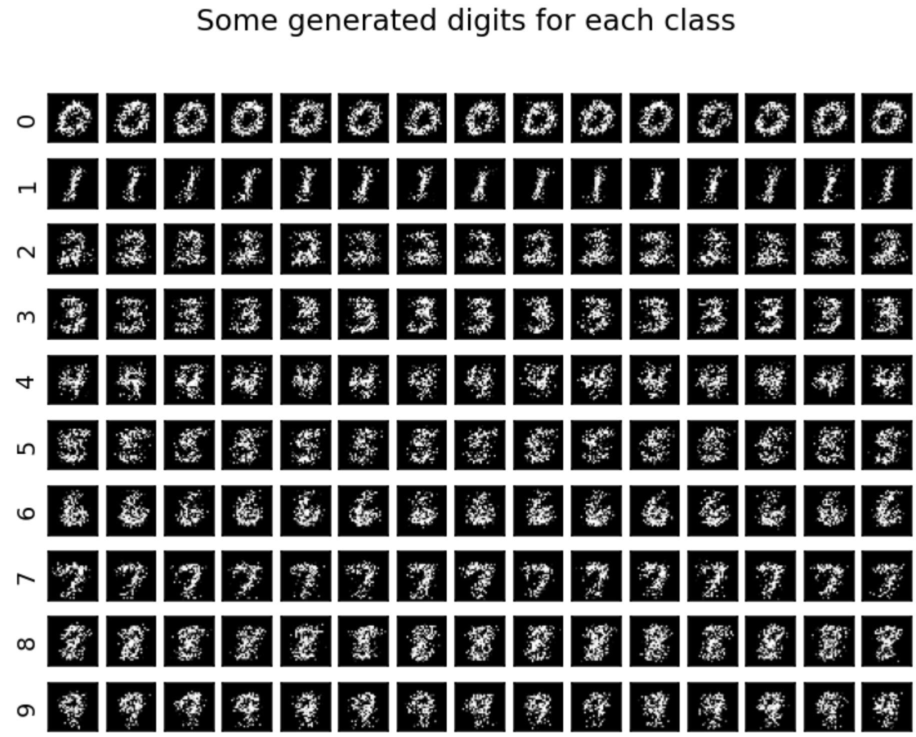


Figure 16: Generated digits for $\alpha = 1$. Notice the noise and lack of structure in certain digits, particularly 5 and 8.

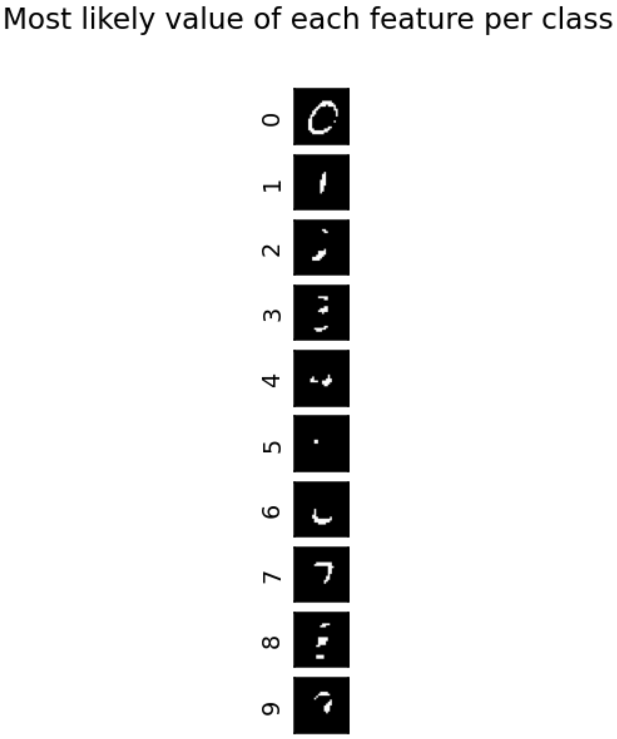


Figure 17: Most likely value for each feature per class. The digits show more noise and distortion, particularly for complex digits like 5 and 8.

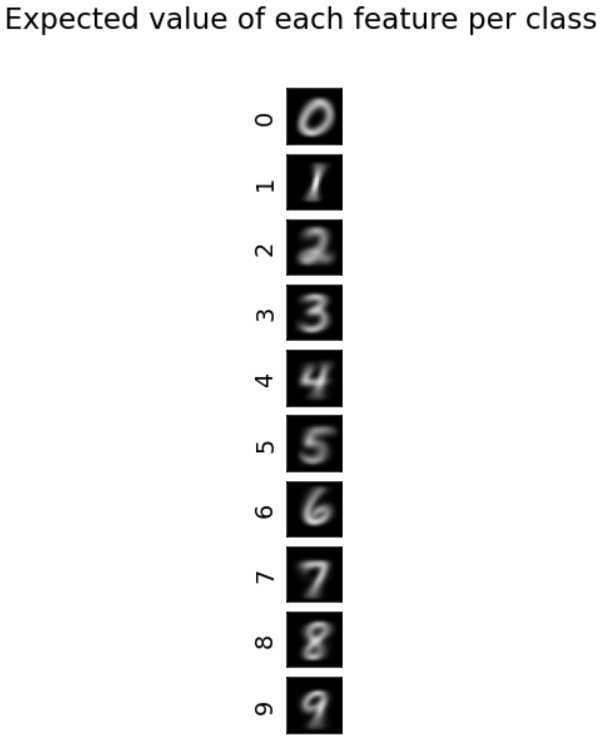


Figure 18: Expected value for each feature per class. The digits are generated less sharp.

D.3. Model with $\alpha = 1.3$

In the following subsection, the model *model_nb3* with $\alpha = 1.3$ is analyzed.

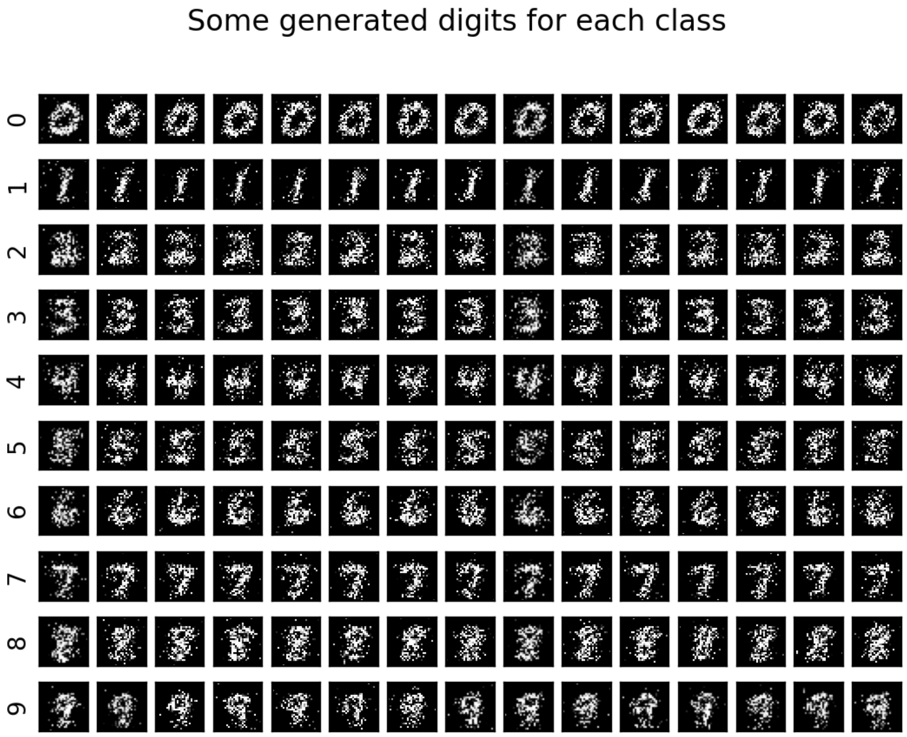


Figure 19: The digits are sharper than those generated with $\alpha = 2$, with minimal noise compared to $\alpha = 1$.

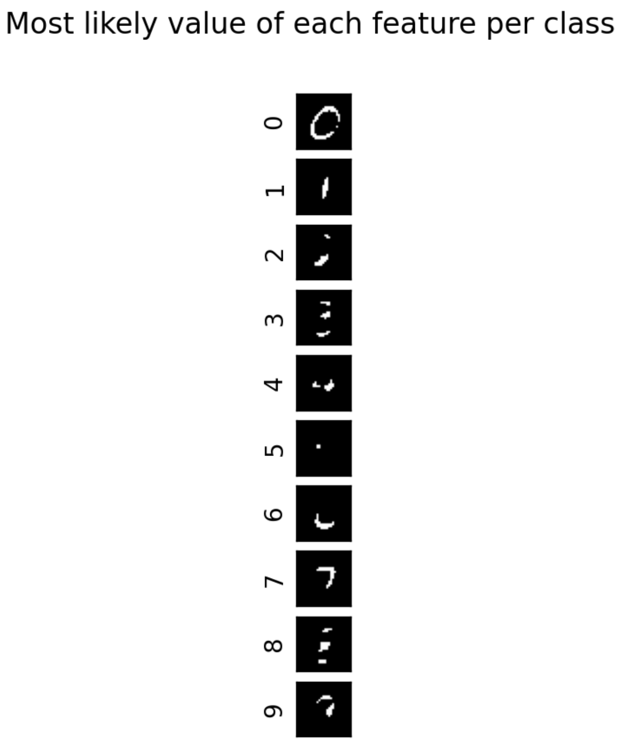


Figure 20: Most likely value for each feature per class for model with $\alpha = 1.3$.

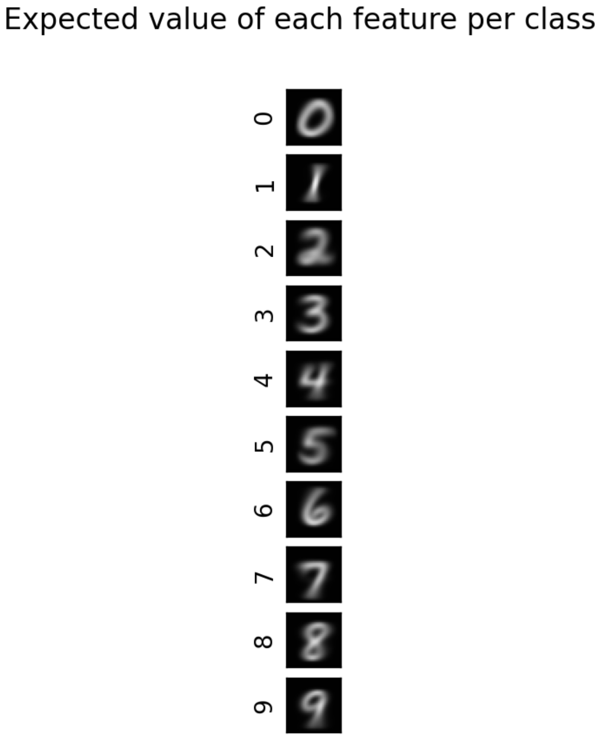


Figure 21: Expected value for each feature per class for model with $\alpha = 1.3$.

Declaration of Honor

I hereby declare that I have written the enclosed project report without the help of third parties and without the use of other sources and aids other than those listed in the table below and that I have identified the passages taken from the sources used verbatim or in terms of content as such. or content taken from the sources used. This work has not been submitted in the same or a similar form been submitted to any examination authority. I am aware that a false declaration declaration will have legal consequences.

Declaration of Used AI Tools

Tool	Purpose	Where?	Useful?
ChatGPT	Rephrasing	Throughout	+
DeepL	Translation	Throughout	+
Github Copilot	Code generation	a01-nb.ipynb	+

Signatures
Mannheim, 12. October 2024

Arne Huckemann
A. Huckemann

Elise Wolff