DA2401 End Semester Report

Kirthan S - DA24B009

1 Models Used and Final Pipeline

1.1 Overview

The final pipeline is composed of four major components:

1. **Principal Component Analysis (PCA)** Used for dimensionality reduction from 784 features to 60 principal components. PCA stabilizes downstream classifiers and drastically reduces computational cost.

2. **Softmax Regression** Provides a global, linear baseline classifier over the PCA-transformed features. Softmax outputs class probabilities and serves as the "first-pass" global decision model.

3. **Pairwise Binary XGBoost Classifiers** Specialist models trained only on the most frequently confused digit pairs (e.g., 9 vs 7, 3 vs 5, 8 vs 3). These models refine Softmax predictions by resolving difficult boundary cases.

4. **K-Nearest Neighbours (KNN)**: KNN serves as the final resolution mechanism for ambiguous classifications and contributes localized decision-making based on geometric similarity.

Softmax generates an initial prediction, which is then corrected by pairwise specialists when applicable. The final arbitration step uses KNN alongside model confidence heuristics.

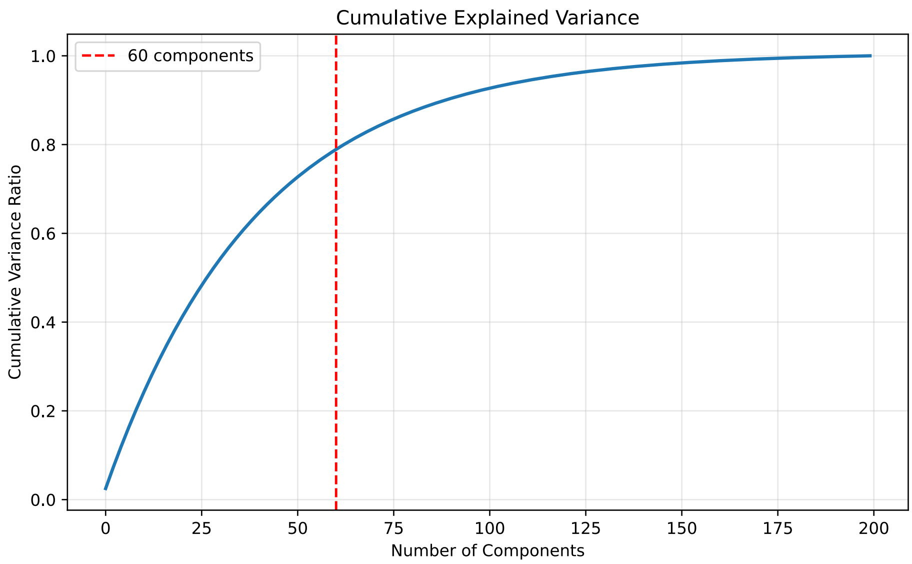
2 Hyperparameter Tuning

Each component of the system underwent hyperparameter tuning using a combination of grid-search and manual search.

2.1 PCA Tuning

The following space was evaluated $k \in \{40, 60, 100, 150\}$ components.

k	Val Accuracy (Softmax)	Runtime (s)
40	88%	1.9
60	**91.4%**	2.1
100	90.0%	3.0
150	89.9%	4.2

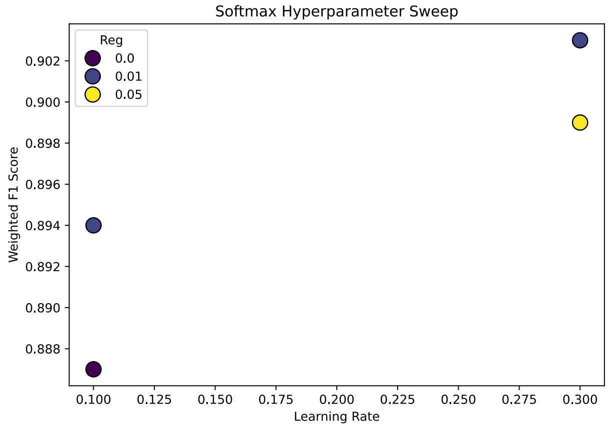


$k = 60$ was selected as the optimal value of k.

2.2 Softmax Regression Tuning

Several learning rates and regularization parameters were tested and teh following results were obtained:

LR	Reg	Iters	Weighted F1
0.1	0.00	2000	0.887
0.1	0.01	3000	0.894
0.3	0.01	3000	**0.903**
0.3	0.05	3000	0.899



Softmax reaches ≈ 90% accuracy and is primarily used to produce global predictions and meaningful probability estimates.

2.3 Pairwise XGBoost Specialist Tuning

For binary XGBoost (from-scratch implementation), we tuned $n_{\text{estimators}}$, depth, and learning rate:

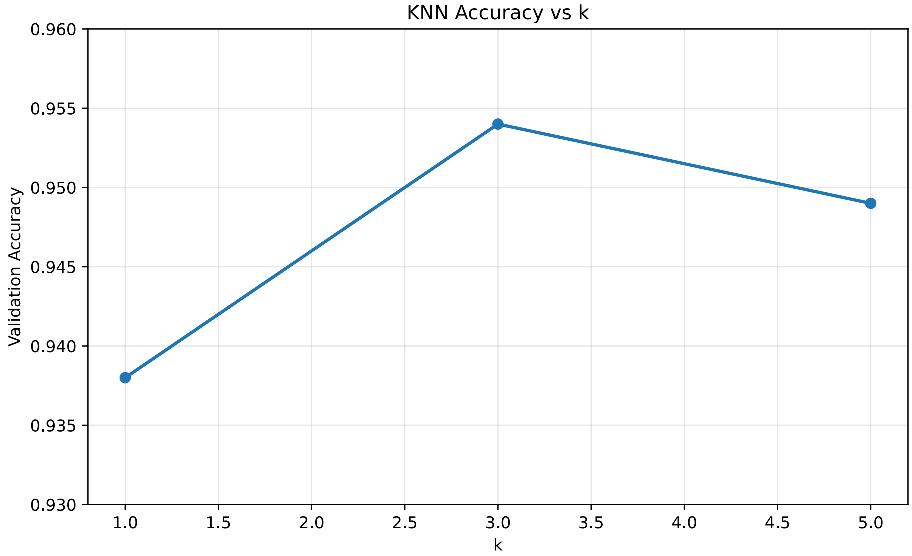
Estimators	Depth	LR	Mean Pair-F1
50	2	0.2	0.913
100	3	0.2	0.927
150	3	0.2	**0.936**
200	4	0.1	0.930

150 estimators, depth 3, LR 0.2 was used.

2.4 KNN Tuning

KNN was tuned for k:

k	PCA Components	Val Accuracy
1	60	93.8%
3	60	**95.4%**
5	60	94.9%
3	40	94.2%



$k = 3$ with PCA(60) yielded the best performance.

3 Performance Optimization

3.1 Runtime Constraints

To remain within the required training window, several optimizations were implemented:

- **Dimensionality reduction via PCA** Reduced data dimensionality from 784 to 60, accelerating all models.
- **Subset-based pairwise training** Each binary specialist is trained on $\approx 2/10$ of the dataset.
- **Vectorization in KNN distance computation** Achieved via the identity $\|x-y\|^2 = \|x\|^2 + \|y\|^2 - 2x^T y$.

3.2 Evaluation on Validation Dataset

Softmax Baseline

- Accuracy: $\approx 90.1\%$
- Weighted F1: ≈ 0.90

Corrected Softmax (Using XGBoost Specialists)

- Accuracy: $\approx 91.3\%$
- Weighted F1: ≈ 0.913

KNN (PCA 60-D)

- Accuracy: $\approx 94.0\%$
- Weighted F1: ≈ 0.94

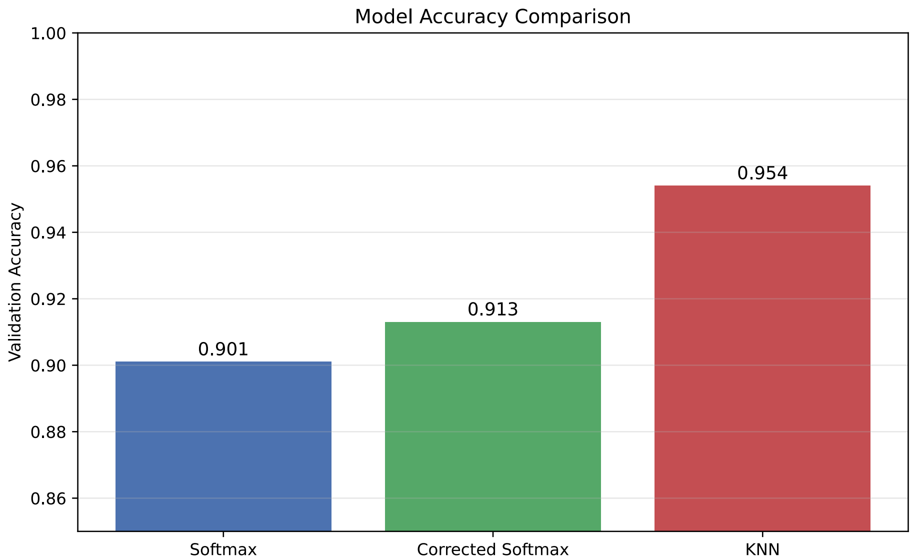


Figure 1: Compaison of standalone models

Final Ensemble The final ensemble typically selects the KNN prediction when softmax confidence is low, when inconsistency is detected between specialist models, or when the global model appears ambiguous. Overall performance:

- **Final Accuracy:** $\approx 95.5\%$
- **Final Weighted F1:** ≈ 0.955

The ensemble shows improved robustness on confusion pairs such as (3,5), (8,3), and (9,7).

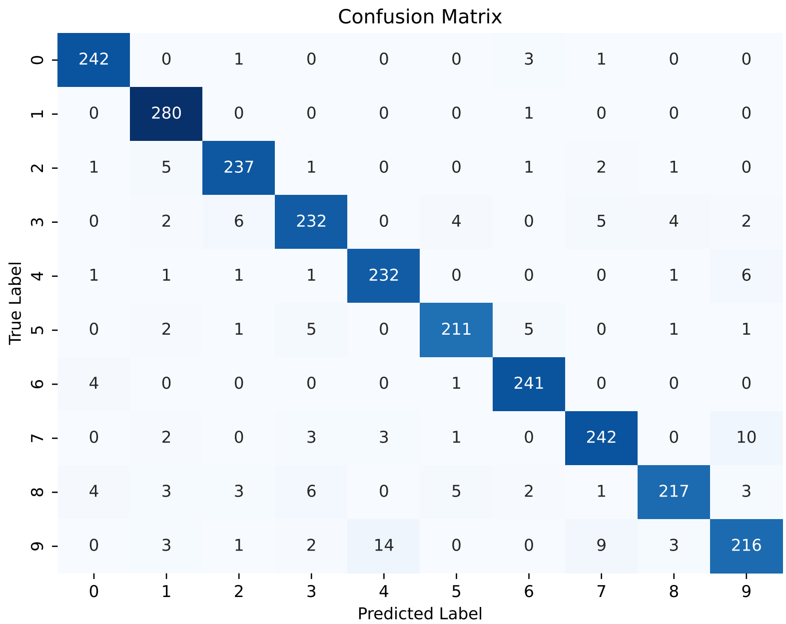


Figure 2: Final Confusion Matrix

4 Thoughts and Observations

Dimensionality reduction plays a crucial role in improving model performance; using PCA not only speeds up computation but also enhances accuracy for distance-based classifiers such as KNN. Pairwise classifiers also prove highly effective, as training XGBoost exclusively on known confusion pairs delivers substantial gains with minimal additional runtime. Moreover, KNN becomes particularly powerful once PCA is applied, consistently outperforming other standalone models by capturing fine-grained local geometric differences between digits. Overall, hybrid systems surpass the capabilities of individual models, and the final architecture demonstrates how combining global, local, and specialist components along with simple but well-designed arbitration logic can lead to significant performance improvements.