# Report

Krish Dange

November 2025

## 1 Summary of Models Used & System Architecture

### 1.1 PCA Preprocessing

- MNIST images have 784 raw pixel features.
- PCA was used to reduce dimensionality.
- Reduced to 100 principal components.
- Shapes after PCA:
  - Train: (10002, 100)
  - Validation: (2499, 100)
- Total variance preserved: 91.56%.
- PCA greatly reduced computation time and improved model stability.

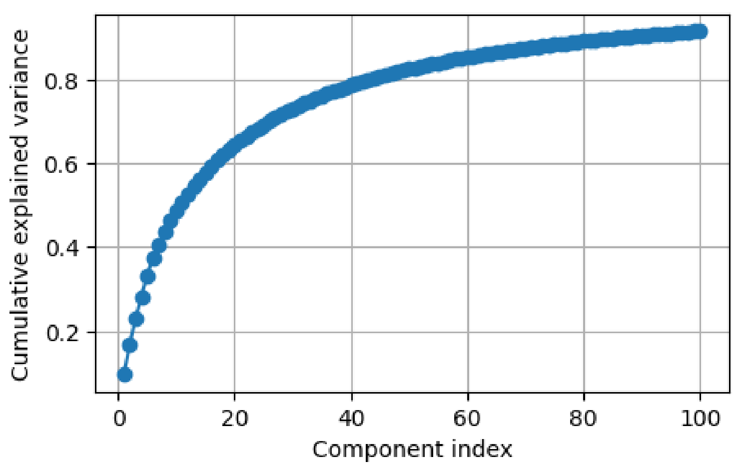


Figure 1: Cumulative explained variance for 100 PCA components

### 1.2 Models Implemented

**Linear Models**

- Softmax Logistic Regression
- Linear SVM (One-vs-Rest)

**Non-linear Models**

- Random Forest
- One-vs-Rest Gradient Boosting

## 1.3 Initial Model Performance

| Model | Weighted F1 | Accuracy | Train Time (s) |
|---|---|---|---|
| Logistic Regression | 0.8561 | 0.8575 | 0.35 |
| Random Forest | 0.7071 | 0.7135 | 97.43 |
| One-vs-Rest GB | 0.5753 | 0.5722 | 19.52 |
| Linear SVM | 0.8645 | 0.8667 | 0.72 |

Observation:

- Linear models clearly outperform scratch non-linear models.
- Thus, only Logistic Regression and SVM were tuned further.

# 2 Hyperparameter Tuning Summary

## 2.1 Softmax Logistic Regression

### 2.1.1 Learning Rate

| Learning Rate | Weighted F1 |
|---|---|
| 0.02 | 0.9032 |
| 0.01 | 0.8944 |
| 0.005 | 0.8898 |

Best: **0.02**

### 2.1.2 Regularization

| reg | Weighted F1 |
|---|---|
| 1e-4 | 0.9041 |
| 1e-5 | 0.9053 |
| 1e-6 | 0.9044 |

Best: **1e-5**

### 2.1.3 Batch Size

| Batch Size | Weighted F1 |
|---|---|
| 128 | 0.9080 |
| 256 | 0.9032 |
| 512 | 0.8964 |

Notes:

- Batch 128 gives highest score but unstable.
- Batch 256 chosen for stability and consistent convergence.

**Final Tuned LR Hyperparameters**

- lr = 0.02
- reg = 1e-5
- batch_size = 256
- epochs = 400
- patience = 30
- Final Weighted F1: **0.905**

## 2.2 Linear SVM (One-vs-Rest)

### 2.2.1 C Tuning

| C | Weighted F1 |
|---|---|
| 10 | 0.8627 |
| 50 | 0.8939 |
| 100 | 0.8980 |
| 300 | 0.9000 |
| 500 | 0.8988 |

Best: **C = 300**. Higher C improves fit until 300, then drops (overfitting).

### 2.2.2 Learning Rate

| Learning Rate | Weighted F1 |
|---|---|
| 0.01 | 0.8964 |
| 0.02 | 0.9000 |
| 0.03 | 0.8988 |

Best: **0.02**

**Final Tuned SVM Hyperparameters**

- C = 300
- lr = 0.02
- batch_size = 256
- epochs = 400
- patience = 30
- Final Weighted F1: **0.900**

# 3 System Optimization Steps & Runtime Performance

- Applied PCA(100) to reduce dimensionality and speed up all models.
- Used efficient NumPy vectorized operations for all model training.
- Mini-batch gradient descent (batch = 256) for stable and fast convergence.
- Learning rate decay: $lr_t = \frac{lr}{1+0.005t}$.
- Early stopping (patience = 30) to prevent unnecessary epochs.
- All models trained under single-core CPU as required.

### Final Validation Results

- Logistic Regression: **F1 = 0.905**, Acc = 0.906
- Linear SVM: **F1 = 0.900**, Acc = 0.899

# 4 Thoughts & Observations

- PCA drastically improved speed and stability; 100 components preserved 91.56% variance.
- After PCA, MNIST became more linearly separable; linear models outperformed non-linear ones.
- Logistic Regression and SVM responded strongly to learning rate and regularization tuning.
- Non-linear scratch models were slower and less accurate, showing the difficulty of manual tree-based implementations.
- Overall best pipeline: **PCA(100) + Softmax Logistic Regression**.
- Training times were extremely low (1–2 seconds), far below the 5-minute constraint.