Math 452: MNIST Midterm Proeject

Dominic Felice

Abstract. This project explores the classification of the handwritten digits in the MNIST dataset using different machine-learning models. These models include K-Nearest Neighbors (KNN), Logistic Regression (LG), and Support Vector Machine (SVM), all provided through the Scikit-Learn package in Python 3. The majority of this project was on classification through model performance with the metrics accuracy, precision, recall, and F1-score. Clustering was also investigated and analyzed using K-Means. To optimize results, cross-validation and hyperparameter tuning were both used with grid search. The highest accuracy achieved by SVM was 96.8%, KNN was 94.9%, and LG was 91.7%. SVM took the longest to train, with KNN being the next longest and LG being the fastest. This is due to the computational intensity of each model and the grid search involved.

In addition to classification, machine learning clustering was also explored. K-means was applied to the data both with and without Principal Component Analysis (PCA) for dimensionality reduction. Despite testing various cluster numbers, the clustering results were limited, and visually the best k value was below 10. When the k value was raised to 10 digit clusters would be repeated. This highlights the challenges that unsupervised learning can cause.

1. Introduction.

This project studies the effectiveness of different machine learning models in attempting to classify handwritten digits using the MNIST dataset. As a school project, the study analyzed the strengths and weaknesses of various algorithms in learning from visual data.

To explore the models' capabilities, three different classification algorithms were used: K-Nearest Neighbors (KNN), Logistic Regression (LG), and Support Vector Machine (SVM). All were accessed through Scikit-Learn in Python3. Each model uses different approaches to learn patterns from data. KNN uses distance metrics, LG uses statistical regression, and SVM optimizes margins for decision boundaries. Additionally, K-Means clustering was implemented to explore unsupervised learning when grouping images. Results were limited due to the challenges of clustering on high-dimensional data.

Our pipeline involved several stages: data normalization, dimensionality reduction, and extensive cross-validation. First, pixel values were normalized and images were flattened, then the dataset was reduced to one-fifth of the original size to reduce computation time. This was done using stratifying to ensure the distribution of data remained the same. KNN was evaluated through cross-validation alone, while hyperparameter tuning using GridSearchCV was used for both LG and SVM. Finally, comparisons across models were performed visually on both individual and aggregate results. For K-Means, configurations were tested both with and without Principal Component Analysis (PCA) for dimensionality reduction, though with limited effectiveness.

2. Related Works.

Handwritten digit classification is a fundamental problem in machine learning, and is often used as a benchmark to evaluate algorithm performance. The MNIST dataset has become one of the standard datasets for this purpose due to its high-dimensional,

structured data which presents challenges to model accuracy and computational efficiency. All types of machine-learning approaches have been employed for MNIST, ranging from classical algorithms like linear regression to new developments in deep-learning methods like neural networks.

In classic machine learning, models like KNN, LG, and SVM have been used based on their interpretability and robust performance. KNN is simple and effective at capturing local patterns, though it can become computationally intensive on large datasets. LG offers a probabilistic approach, making it a good baseline for models with linear decision boundaries. SVM is known for its margin-based optimization, which allows for higher accuracy however it causes greater computational resources.

Clustering with K-Means is a common unsupervised approach to group digits without the use of labels. However, due to the high-dimensional nature of image data, particularly with MNIST, K-Means clustering can struggle without dimensionality reduction. Techniques like PCA are often used to address these dimensionality problems by reducing the complexity of the data before clustering.

This project draws from these established approaches to asses classification and clustering effectiveness on MNIST, with an additional focus on cross-validation and hyperparameter tuning. While most recent studies apply the use of different neural networks, classical methods like KNN, LG, and SVM remain important for their interpretability and reduced complexity,

3. Data. The MNIST dataset is widely used as a benchmark for image classification tasks. It consists of grayscale handwritten digit images, from 0 to 9, each standardized at a 28x28 resolution. Each image's pixel intensity ranges from 0-55, representing the darkness of each of the pixels. By default, 60,000 training images and 10,000 test images.

3.1. Preprocessing and Data Reduction. To prepare the data for model training, the pixel values were normalized from 0 to 1 ensuring consistent feature scaling across the models. Each image was then flattened to a 784-dimensional vector, transforming the two-dimensional structure into a one-dimensional structure, a more suitable format for machine learning.

To reduce computational demands while maintaining class distribution, stratified sampling was employed to select one-fifth of the original data. Because of this, 12,00 training images and 2,000 images were set, corresponding to an approximate 85/15 training-testing split. The stratification preserves the original data distribution, ensuring a balanced representation of each digit class in both training and testing sets.

3.2. Dimensionality Reduction for Clustering. In addition to flattening, PCA was also applied to the reduced data for the K-Means Model. PCA helped to reduce the dimensionality even further, allowing for more efficient clustering, despite how limited the outcomes were due to the complex structure of handwritten, visual data.

4. Methods. In this project, three classical algorithms were use for classification, K-Nearest Neighbors (KNN), Logistic Regression (LG), and Support Vector Machine (SVM).

Each of these algorithms represents a different approach to machine learning. KNN is an instance-based learner that uses similarity or distance metrics, making it good at capturing local patterns in the data. LG serves as a reliable baseline for establishing linear decision boundaries. SVM is more effective at handling higher-dimensional spaces through margin-based optimization. The models were selected for their effectiveness as foundational algorithms in machine learning classification tasks.

4.1. K-Nearest Neighbors. To identify the best number of neighbors (k) for KNN, an extensive cross-validation process was employed. A 10-fold stratified cross-validation was repeated 10 times while using different random seeds to ensure the robustness of the results. K values of 1 to 10 were tested while recording their accuracy scores across all folds and repetitions. This allows for accountability of variability and ensuring that the chosen k value generalizes well across different data splits.

4.2. Logistic Regression. For LG, hyperparameter tuning was used with GridSearchCv to optimize the regularization parameter and its penalty. A parameter grid was established with regularized and unregularized settings with an L2 penalty to explore a broad range of methodologies. Specifically, the solver Limited-memory Broyden-Fletcher-Goldfarb-Shanno (lbgfs) was tested with an L2 penalty at .05, .1, 1, and 10 regularization parameters. After that, lbfgs was tested with no penalty and no solver. A five-fold cross-validation was used inside of the grid search, further maximizing the model's accuracy.

4.3. Support Vector Machine. SVM was also fine-tuned using GirdSerachCV, with an established parameter grid that included different kernel types and regularization parameters. Specifically, the kernels Polynomial (poly), Radial Basis Function (rbf), and linear were tested at regularization values of .1, 1, and 10. This setup allowed the evaluation of different margin-maximizing strategies across five-fold cross-validation.

4.4. K-Means Clustering. For unsupervised clustering, K-Means was applied with and without PCA to assess its efficiency on high-dimension data. Cluster values (k) were tested for the range of 1 to 15. The non-PCA was performed directly on the flattened data, while the PCA reduced the data to five principal components. Within-cluster sum of squares (WCSS) was calculated to evaluate clustering compactness and determine the optimal number of ks.

5. Experiments. Experiments were done on a local machine using an Intel i9-10980HK processor with 32GB of DDR4 3200MT/s ram. Although computational resources were not a limiting factor for this project, training time and computational efficiency of models were significant in evaluating their effectiveness.

5.1. Evaluation Metrics. For classification models, accuracy, precision, recall, and F1-score were the primary evaluation metrics. All these metrics provide a comprehensive view of each model's performance. Training time also came into play to track each model's computational efficiency. Additionally, ROC curves and the percentage of times each digit was classified correctly were looked at.

5.2. K-Nearest Neighbors. After extensive cross-validation was conducted, the optimal value k for KNN was found to be 1 at a testing accuracy of 94.9% with a training time of 3 minutes and 7.3 seconds.

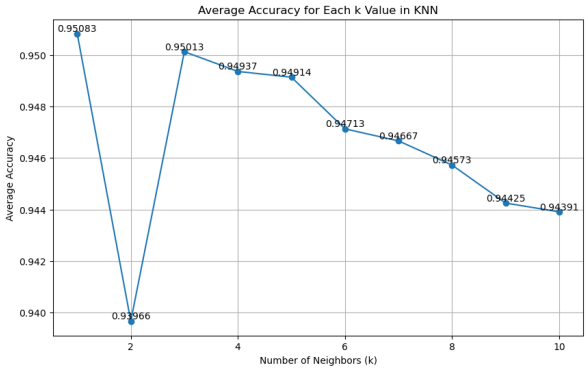

Figure 1. *Average accuracy of KNN across folds*

Additionally, the ROC curve plot shows that the AUC values for all classes were quite high, ranging from .95 to .99. This indicates the k-value of 1 on the test data performs well in distinguishing each digit class from the others.

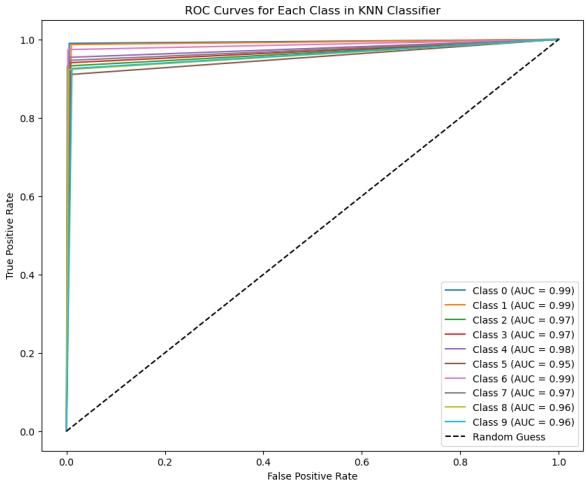

Figure 2. *ROC Curves of KNN for k=1*

5.3. Logistic Regression. After hyperparameter tuning via GridSearchCV, Logistic Regression's best accuracy was 91.7% using lbfgs with an L2 parameter at 0.1 The training time for this model was 1 minute and 6 seconds.

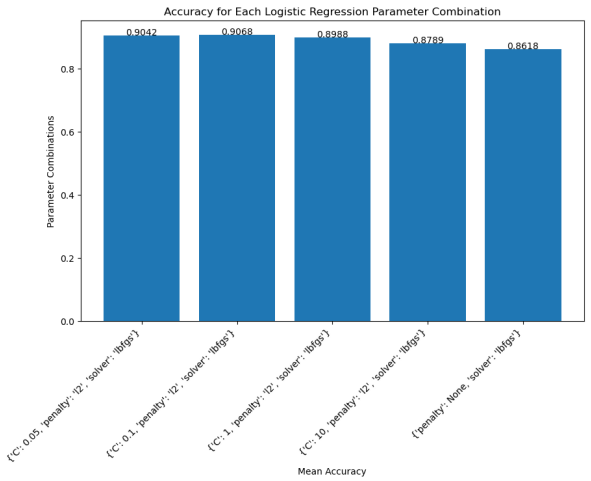

Figure 3. *Logstic Regression Accuracies*

While LG also had high values of AUC ranging from .92 to .99, they were not as good on average as KNN showing that it is less confident at distinguishing between classes.

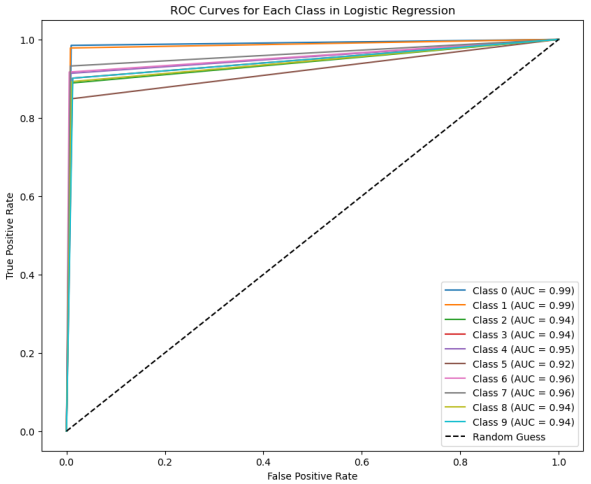

Figure 4. *Logistic Regression ROC*

5.4. Support Vector Machine. SVM had the highest accuracy at 96.8% after extensive tuning of kernel types and regularization parameters. The best parameters found were kernel rbf with a regularization of 10. However, SVM's training time was longer than all the others at 12 minutes and 25 seconds, further demonstrating the trade-off between accuracy and computational efficiency.

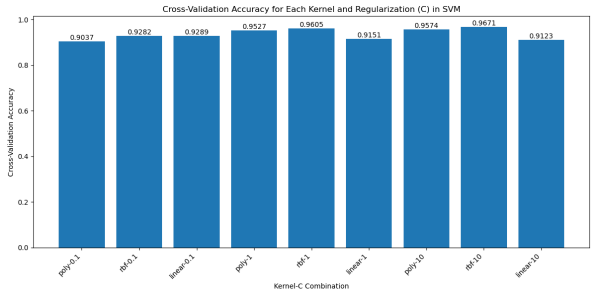

Figure 5. *Support Vector Machine Accuracies*

Additionally, SVM had the best on-average AUC values, demonstrating that it is very confident in distinguishing between the classes.

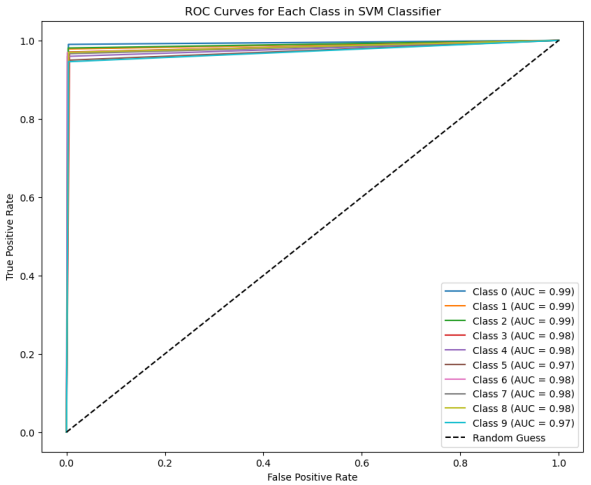

Figure 6. *Support Vector Machine ROC*

5.5. K-Means Clustering. For unsupervised learning, K-Means clustering was used to assess its ability for grouping digits without any label data. The optimal cluster was determined using a visual method, the elbow method, and identified 5 clusters for the PCA-reduced data and 6 clusters for the non-PCA dat as optimal. With six clusters. For non-PCA, K-means it appears that the digit 9 was repeated. With 10 clusters (the total number of digits), 9 and 1 were both repeated twice with 8 and 4 being non-present in the results.

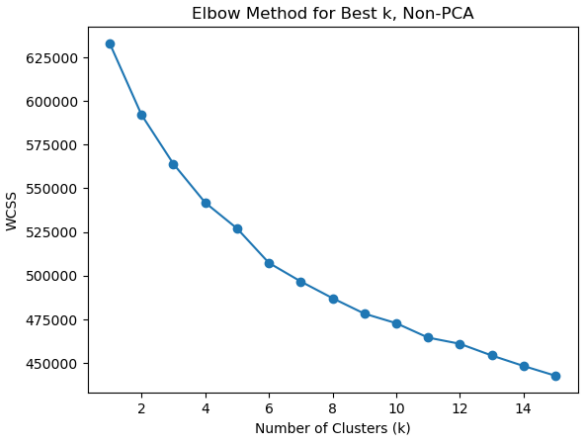

Figure 7. *Elbow Graph for Non-PCA*

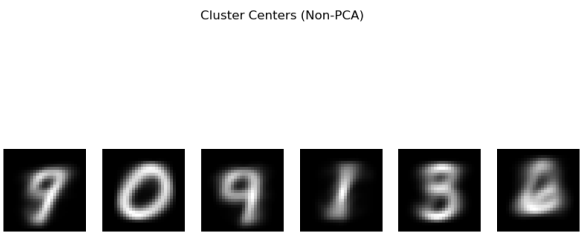

Figure 8. *Non-PCA Best Clusters*

Figure 9. *Non-PCA Experimental Cluster*

For PCA K-Means, it was hard to distinguish what digits were present with the clustering of five. However, they appear to be 1, 9 6, 3, and 0. When 10 clusters are selected it appears 3 was repeated twice and 0 and 9 were repeated three times, with the digits 2, 4, 5, 7, and 8 not being in the results.

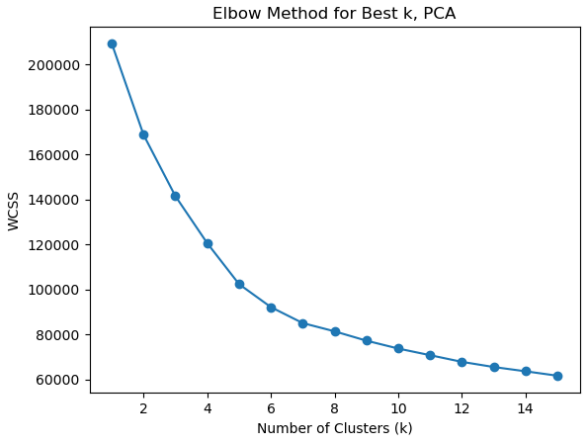

Figure 10. *PCA Elbow Graph*

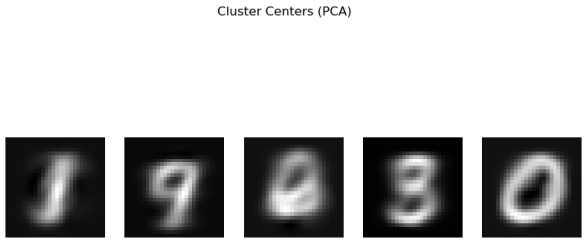

Figure 11. *PCA Best Cluster*

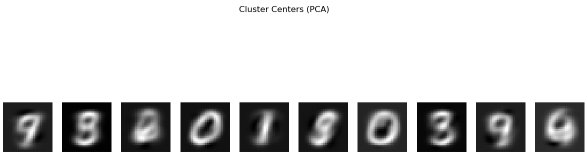

Figure 12. *PCA Experimental Cluster*

While these results provide some initial structure, the limitations of clustering with unsupervised learning are evident. The complex nature of handwritten digits shows how K-Means can struggle to separate digit classes effectively, often grouping similar-looking digits and repeating digit classes within clusters.

5.6. Classification Model Comparisons. To evaluate the performance of each classification model, accuracy, precision, recall, and F1-Score were all compared across KNN, LG, and SVm. Notably, the accuracy, precision, recall, and F1-score were nearly identical for each respective model. This shows that the models were very balanced in their classification performance and suggest that they were consistent in identifying true positives and minimizing false positives, without any major biases.

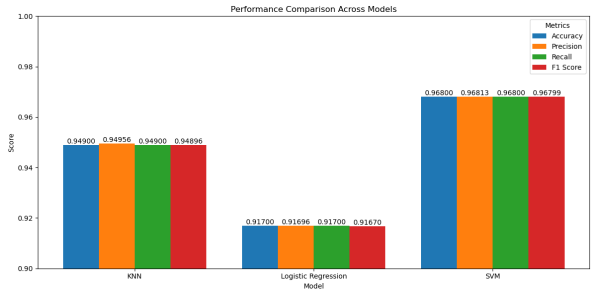

Figure 13. *Metrics of Each Model*

Across the models, certain patterns emerged in classifying the digits. For example, the digits 5, 8, and 9 were on average the most misclassified digits, suggesting that these digits share visual characteristics. On the other hand, the digits 0 and 1 were on average accurately classified the most, suggesting that they have distinct shapes that are easy for the model to distinguish between.

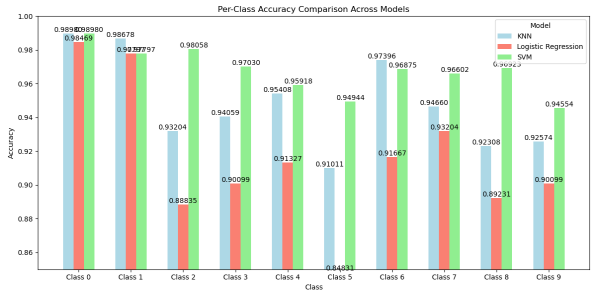

Figure 14. *Per-Model Accuracy of Each Class*

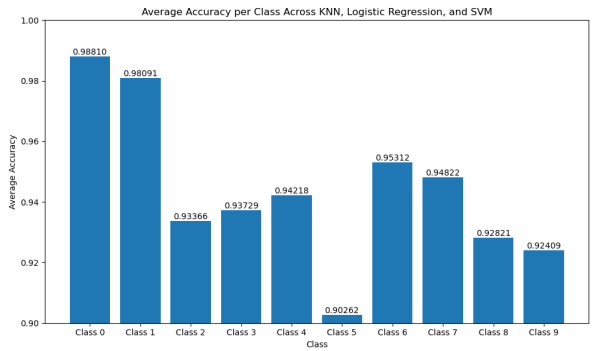

Figure 15. *Average Accuracy of Models for each Class*

6. Conclusion. This project explored the effectiveness of three classification machine-learning algorithms, Knn, LR, and SVM, while also taking an experimental glance at clustering through K-Means on the MNIST dataset.

Among the classification models, SVM was the most accurate at 96.8% accuracy followed by KNN at 94.9% and LG at 91.7%. This shows SVM's strengths in high-dimensional data, especially when tuned properly with different kernels and regularization parameters. However, SVM was also computationally more intensive, causing it to have significantly longer running times than the other two algorithms. This highlights the trade-off between accuracy and computational efficiency in machine-learning, a subject that is always being studied and expanded upon. KNN provided a good balance between LR and SVM, running at a faster time than SVM while being more accurate than LR.

Clustering with K-means, with and without PCA showed limited effectiveness in this project. The optimal number of clusters for both was significantly less than the number of classes in the dataset. Even when the number of clusters was increased, the model still struggled to identify unique digits. This could indicate that the model alone is insufficient for capturing complex patterns in handwritten, visual data. In order to further increase performance, better pre-processing or other means would need to be used. This underscores the challenges of using unsupervised learning techniques, both in this context and in others.

In summary, this projects highlights the strengths and weaknesses of different Machine learning models when used on high-dimensional, handwritten, visual data. LG was the fastest but least accurate, SVM was the slowest but most accurate, and KNN was in between them for both metrics. Clustering with K-Means, while insightful, demonstrated the lack of utility for certain algorithms in certain contexts, and highlights the need to always explore multiple models and ensure proper data preprocessing. Future work could other dimensionality reduction techniques for K-Means, the use of deep or reinforcement learning models, and further hyperparameter tuning for all models in dataset to improve performance.