Assignment 2

Logistic Regression and SVM.

David Padilla Orenga
Ignacio Pastore Benaim

October 4, 2024

1 Introduction

In this report, we present the results of the second assignment of the Machine Learning course. The objective of this assignment is to apply Logistic Regression(LR) and Support Vector Machines (SVM) on the MNIST dataset.

The MNIST dataset consists of grayscale images of handwritten digits ranging from 0 to 9, with 784 features representing pixel intensities. Although the primary focus was on Logistic Regression and SVM, we extended our exploration to include a range of additional models. These include a k-Nearest Neighbors (k-NN) classifier for its simplicity, a Ridge Classifier (RG) to test regularization in a classification setting, and finally, a Voting Classifier (VC) ensemble to combine the strengths of our top-performing models.

The analysis consists of three key phases: initial validation, cross-validation, and final testing. Finally, a short discussion and conclusion are presented.

The complete code used for this analysis can be found in a Google Colab Notebook.

2 Materials & Methods

The MNIST dataset comprises 70,000 grayscale images of handwritten digits, each having 784 features corresponding to pixel intensities. The dataset was split into training (81%), validation (9%), and test (10%) sets. In order to speed up convergence of the models, first we scale the data using the StandardScaler, and afterwards we applied a Principal Component Analysis (PCA) in some evaluation phases.

All models were implemented in a local machine with Python using the numpy, pandas, and scikit-learn libraries.

2.1 Initial Screening with Validation Set

Various models (detailed in Section 2.4) were evaluated initially using the validation set. This phase aimed to identify suitable candidates and establish a baseline performance. The models were tested with and without dimensionality reduction using PCA.

2.2 Cross-Validation of Selected Models

Based on the initial validation results, a subset of models was chosen for cross-validation using $k = 5$ folds. The cross-validation was conducted with 20 PCA components and evaluated with the training and validation sets together.

2.3 Final Testing on Test Set

After cross-validation, the optimal models were retrained using the combined training and validation set. The final evaluation was performed on the independent test set, and the results were used to assess the models' generalization capabilities.

2.4 Models

For each model, we specify whether Principal Component Analysis (PCA) was applied and if hyperparameter optimization was performed using GridSearch.

2.4.1 Logistic Regression

LR was tested with and without PCA over 20, 30, 40, 50 and 100 components.

2.4.2 k-Nearest Neighbors (k-NN)

k-NN was tested with and without PCA over 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500 and 600 components. After observing that with 20 PCA components and $k = 1$ the model retained around 95% of accuracy, we conducted a GridSearch with PCA over the number of neighbors k, ranging from 1 to 20.

2.4.3 Ridge Classifier

A GridSearch was performed over the regularization parameter λ but did not apply PCA, as the model was discarded early due to suboptimal performance.

2.4.4 Support Vector Machine (SVM)

Given the computational expense of SVMs, PCA was applied with 20 components. A GridSearch was attempted for the regularization parameter C, kernel types and kernel parameters (degree, gamma). However, due to computational limitations, we focused on a Radial Basis Function (RBF) kernel with the regularization parameter $C = 1$ and the default $\gamma =$ 'scale'.

The default value 'scale' uses the formula:

$$\gamma = \frac{1}{n_{\text{features}} \cdot \text{Var}(X)}$$

where n_{features} is the number of features and $\text{Var}(X)$ is the variance of the input data.

2.4.5 Voting Classifier

To leverage the strengths of individual models, a Voting Classifier was implemented, combining the top-performing models.

2.5 Evaluation Metrics

Various metrics were employed during different phases of the analysis.

- **Accuracy:** The proportion of correctly classified instances.
- **Precision:** The proportion of correctly classified positive instances among all instances classified as positive.
- **Recall:** The proportion of correctly classified positive instances among all actual positive instances.
- **F1-Score:** The harmonic mean of precision and recall.
- **Time:** Tracking of time in different tasks.

3 Results

3.1 Initial Validation Results

The initial screening of models was conducted using the validation set. Table 1 shows the accuracy scores for each model with and without PCA and the time spended in training and predicting.

Given these results, the Ridge Classifier was discarded for further analysis. The remaining models were selected for cross-validation with 20 PCA components in search of optimality between performance and computational workload.

3.2 Cross-Validation Results

The cross-validation results are shown in Table 2. The models were evaluated using 5-fold cross-validation with 20 PCA components. The Logistic Regression model was discarded due to its suboptimal performance.

Table 1: Initial validation performance. Time: elpased time of training + predicting.

Model	Accuracy	Time(s)
k-NN (k=6), PCA=20	0.953	0.445
k-NN (k=6), PCA=30	0.958	0.473
k-NN (k=6), PCA=40	0.960	0.590
k-NN (k=6), PCA=50	0.962	0.609
k-NN (k=6), PCA=100	0.961	0.713
k-NN (k=6) (All dims)	0.945	3.284
LR (PCA=20)	0.871	3.621
LR (PCA=30)	0.885	3.110
LR (PCA=40)	0.895	3.593
LR (PCA=50)	0.900	3.803
LR (PCA=100)	0.913	8.523
RC ($\lambda = 0.001$)	0.771	2.361
RC ($\lambda = 0.01$)	0.771	4.507
RC ($\lambda = 0.1$)	0.771	6.423
RC ($\lambda = 1$)	0.771	8.806
RC ($\lambda = 10$)	0.771	10.704
RC ($\lambda = 100$)	0.771	12.105
SVM (PCA=20)	0.961	27.730
SVM (PCA=30)	0.967	30.754
SVM (PCA=40)	0.970	36.696
SVM (PCA=50)	0.971	44.384
SVM (PCA=100)	0.971	88.341

Table 2: Initial validation performance. Mean Accuracy over 5 folds. Time: elapsed time of training + cross-validating.

Model	Mean Acc.	Time(s)
k-NN (k=6), PCA=20	0.947	2.556
LR, PCA=20	0.868	17.605
SVM, PCA=20	0.961	168.414
VC (LR + k-NN + SVM)	0.951	190.714

Table 3: Final testing. F1, Recall and Precision are the weighted average for all labels. Time: elapsed time in training and predicting.

Model	F1	Recall	Precision	Time(s)
k-NN (k=6),	0.948	0.948	0.948	0.482
SVM (C=1; kernel=rbf),	0.960	0.960	0.960	33.174
VC (k-NN + SVM)	0.951	0.951	0.951	33.642

3.3 Final Testing Results

The final testing results are shown in Table 3, Where F1, Recall and Precision are the weighted average for all labels. The models were retrained using 20 PCA components on the combined training and validation set and evaluated on the test set. K-NN and SVM were implemented with the same hyperparameters as in the cross-validation phase. VC was implemented with the k-NN and SVM, leaving out the LR model.

4 Discussion

The results show that Support Vector Machines (SVM) achieved the highest performance across all evaluation metrics, followed closely by the Voting Classifier (VC) and k-Nearest Neighbors (k-NN). Logistic Regression (LR), despite being conceptually simple and computationally efficient, performed poorly in comparison. This can be attributed to the linear nature of LR, which limits its ability to capture complex patterns in high-dimensional data like MNIST. Consequently, LR was excluded from the final Voting Classifier due to its comparatively lower accuracy and computational cost.

K-NN, on the other hand, demonstrated strong performance. Additionally, the dimensionality reduction significantly decreased the computational time.

SVM, showed consistent high accuracy with 20 PCA components. However, the computational time was considerably higher compared to k-NN. This trade-off between accuracy and computational cost was one of the key considerations in model selection.

The Voting Classifier, combining k-NN and SVM, achieved a balanced performance with high accuracy and reasonable training time. The ensemble's performance indicates that combining complementary models can enhance the robustness and overall performance of the classifier, making it a strong candidate for deployment.

5 Conclusion

Based on the final evaluation results, the Support Vector Machine (SVM) emerged as the best-performing model, achieving the highest scores on the test set. However, the high computational cost of SVM makes it less desirable for real-time or resource-constrained environments. k-NN, while less computationally intensive, achieved comparable performance and proved to be a simpler yet effective choice.

The Voting Classifier, which combined k-NN and SVM, offers a good balance between performance and computational efficiency. Although it did not surpass SVM in individual metrics, its robustness and ease of implementation make it an attractive option.

The final choice of model will depend on the specific requirements of the application. For resource-constrained environments, k-NN may be the preferred choice due to its simplicity and efficiency. However, for applications where accuracy is preferred over speed, SVM or the Voting Classifier may be more suitable.