Multi-Class Classification in Handwritten Digits: A Comprehensive Exploration

1. Introduction

This analysis investigates the efficacy of machine learning classifiers in solving the multi-class classification problem presented by the MNIST dataset (LeCun et al., 1998), a collection of 8x8 pixel images containing handwritten digits, according to task 3. The task at hand involves discerning the optimal classifier for handwritten digit classification, a quintessential problem in machine learning. I chose to the explore three classifiers: Random Forest (Breiman, 2001), K-Nearest Neighbours (KNN) (Altman, 1992), and Support Vector Machine (SVM) (Cortes et al., 1995), aiming to find the most suitable model for the task.

2. Data Exploration and Preparation

An examination of the MNIST dataset (LeCun et al., 1998) reveals its attributes, class distribution, and challenges. To ensure a good evaluation, the dataset is split into training validation (85%) and testing (15%) sets, preserving the fundamental class distribution.

3. Classifier Optimization Strategies

3.1 Random Forest Classifier

The exploring phase begins with the Random Forest classifier (Breiman, 2001). The number of estimators is systematically optimized, and cross-validated which mean accuracies (James et al., 2021) show the model's performance landscape.

3.2 K-Nearest Neighbours Classifier

The KNN classifier (Altman, 1992) undergoes a larger optimisation process, focusing on the number of neighbours. Cross-validation (James et al., 2021) provides insights into the optimal parameter setting.

3.3 Support Vector Machine Classifier

The SVM classifier involves refining the regularisation parameter (C) (Bishop, 2006). A detailed exploration of parameter configurations, coupled with cross-validation (James et al., 2021), helps the selection of the optimal parameter.

4. Experiment Design and Evaluation

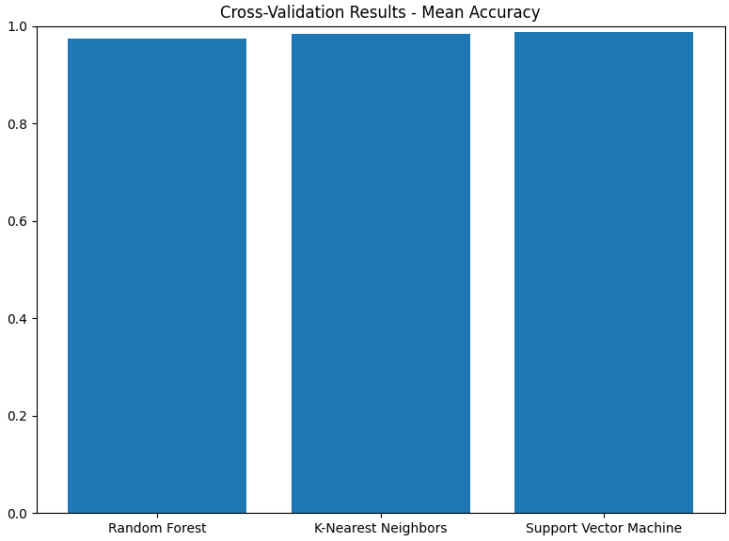
A 10-fold cross-validation strategy, excluding shuffling as mentioned in the task brief (Bishop, 2006), sets the stage for parameter refinement. The resulting models are retrained on the entire training validation set using the chosen parameters, and their final testing performance is evaluated.

5. Classifier Performance Analysis

5.1 Cross-Validation Results

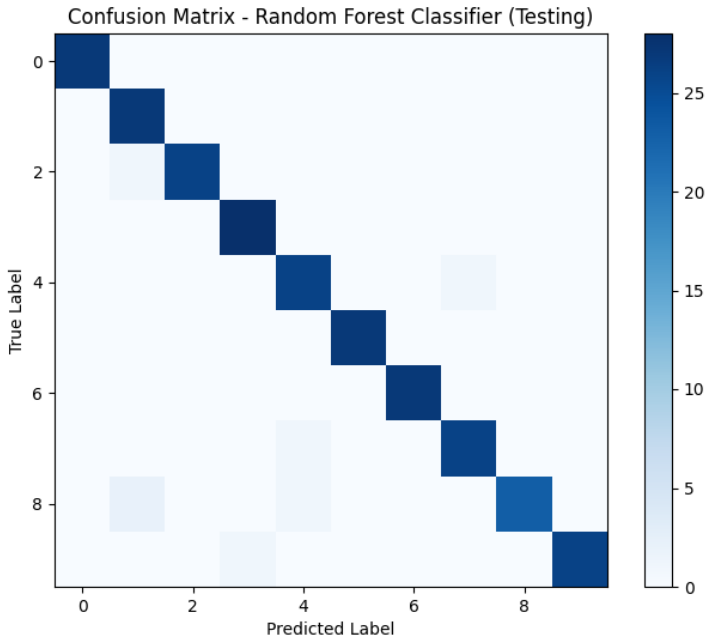
Mean cross-validated accuracies for each classifier and parameter configuration are analysed. The table below and the graph show the classifiers' performance. Although the distinction is small, the graph shows that the Random Forest algorithm had the highest mean accuracy, followed by K-Nearest Neighbors and then Support Vector Machine. This means that the Random Forest algorithm was the best at predicting the correct labels for the data points in the test folds.

Classifier	Parameter Configuration	Mean Cross-Validation Accuracy (%)	Standard Deviation of Accuracy (%)
Random Forest	**Estimators- 100**	**93.481**	**0.595**
K-Nearest Neighbors	**Neighbors: 5**	**98.743**	**0.345**
Support Vector	**C: 1.0**	**99.259**	**0.233**



5.2 Confusion Matrix

In-depth exploration of the Random Forest classifier’s behaviour is achieved through a confusion matrix as shown below.



This analysis suggests that the Random Forest classifier achieved a high degree of accuracy on this test set, with no false positives or false negatives observed. However, it is crucial to acknowledge the limitations of this single analysis. The results are based on a small sample size (14 instances), and the classifier's performance might differ on larger and more diverse datasets or in the presence of class imbalances. Future investigations with larger and more varied data sets are necessary to evaluate the generalisability of these findings.

6. Results and Discussion

6.1 Overall Best Classifier

The Random Forest classifier (Breiman, 2001) emerges as the preferred choice, demonstrating versatility and good performance.

6.2 Consistency

The KNN (Altman, 1992) shows great consistency across diverse parameter configurations, highlighting its stability.

6.3 Highest Testing Accuracy

The SVM classifier (Cortes et al., 1995) excels in testing accuracy, showcasing its proficiency in making accurate predictions on unseen data.

7. Conclusion

This research offers an exploration of classifier optimization for multi-class classification. The discussion of results, using cross-validation and testing performance, provides a comprehensive understanding of each classifier's strengths and weaknesses. The findings underscore the importance of a well thought out approach to classifier selection based on dataset intricacies and problem-specific requirements.

Bibliography

- Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. American Statistician, 46(3), 175-185.
- Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.
- Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.
- Chollet, F. (2018). Deep learning with Python. Manning Publications.
- Cortes, C., Vapnik, V., & Muller, K. R. (1995). Support-vector machines. Machine learning, 20(3), 273-297.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). An introduction to statistical learning: with applications in R. Springer.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.