CMP-6002A Part 3: Optimization and Evaluation of Classifiers

John Callaghan

registration: 100384938

1 Introduction

The main goal of this project is to assess the performance of supervised learning algorithms on the MNIST dataset, Using 3 separate supervised classifiers being Random Forest, K-Nearest Neighbor (K-NN), and Support Vector Machine classifiers. Each classifier is optimized using 5-fold cross-validation to evaluate their performance before and after hyper parameter tuning.

2 Methods

2.1 Data Preparation

The MNIST dataset which is a dataset consisting of greyscale images samples each with 8x8 pixels of resolution, and the samples are fairly distributed across ten classes (0-9). This was then split 85/15 for training and testing along with using 5-fold cross-validation without shuffling, The mean accuracy and the standard deviation of each model was then calculated using default hyper parameters to confirm a baseline for the classifiers before optimization. Figure 1 shows the baseline performance of each classifier, and also shows SVM to be the most accurate of the three models However, the KNN classifier has the smallest standard deviation, meaning it performed the most consistently across the cross-validation folds. This suggests that while SVM may yield the highest overall accuracy, KNN offers a more stable performance.

```
Default Classifier Performance:
                      Classifier  Mean Accuracy  Standard Deviation
0        Random Forest (Default)       0.973800            0.005882
1  K-Nearest Neighbors (Default)       0.988222            0.004402
2                    SVC (Default)       0.984944            0.005305
```

Figure 1: Baseline Performance of Classifiers on MNIST Dataset

2.2 Optimization

Each classifier was optimized by adjusting the relevant hyperparameters to find which works best with the dataset:

- **Random Forest:** Varying ‘n_estimators‘ [100, 150, 200, 300] and ‘max_depth‘ [5, 10, 15, 20, 30].
- **K-Nearest Neighbor:** Testing values of ‘k‘ from 2 to 7.

- **Support Vector Machine:** Varying ‘Kernel‘ [’rbf’, ’poly’] and ‘gamma values‘ [0.001, 0.01, 0.1, 1, 10].

Each cross-validation accuracy score was recorded, with mean accuracy and standard deviation calculated across folds to identify the optimal settings for each classifier.

3 Classifiers Results

3.1 Random Forest

The Random Forest Classifier, which combines multiple decision trees for improved accuracy, was tuned by varying the number of estimators and the maximum tree depth. Optimal Random Forest Configuration: n_estimators: 200 and a max_depth of 15 giving an accuracy of 0.979702 (Shown in Figure 12)

Random Forest Optimization Results:

	n_estimators	max_depth	Mean Accuracy	Standard Deviation
0	100	5	0.939762	0.005185
1	100	10	0.975129	0.007857
2	100	15	0.973800	0.005882
3	100	20	0.973800	0.005882
4	100	30	0.973800	0.005882
5	150	5	0.941733	0.010736
6	150	10	0.975772	0.002598
7	150	15	0.977735	0.006685
8	150	20	0.977735	0.005243
9	150	30	0.977735	0.005243
10	200	5	0.939775	0.009212
11	200	10	0.975772	0.003918
12	200	15	0.979702	0.003803
13	200	20	0.978391	0.004439
14	200	30	0.978391	0.004439
15	300	5	0.943705	0.010061
16	300	10	0.978391	0.004439
17	300	15	0.977083	0.005058
18	300	20	0.976432	0.004298
19	300	30	0.976432	0.004298

Figure 2: Optimized Results for Random Forest Classifier

3.2 K-Nearest Neighbor

The K-Nearest Neighbor algorithm labels a data point based on its closest neighboring values, with the number of neighbors (‘k‘) being a critical hyper parameter for tuning.Steinbach(2024)

Optimal K-Nearest Neighbor Configuration: K value of : 2 giving an accuracy of: 0.989521 (Shown in Figure 9)

```
KNN Optimization Results:
    K  Mean Accuracy  Standard Deviation
0  2       0.989521            0.006359
1  3       0.987567            0.005608
2  4       0.986915            0.007142
3  5       0.988222            0.004402
4  6       0.983628            0.008798
5  7       0.986251            0.006349
```

Figure 3: Optimized Results for K-Nearest Neighbor

3.3 Support Vector Machine

Support Vector Classifiers, which utilize hyperplanes to separate data into different categories Testas A(2023), were optimized by tuning the kernel and gamma parameters.

Optimal Support Vector Machine Configuration: Kernel: 'rbf' Gamma value: 0.001 giving an accuracy of 0.990837 (Shown in Figure 10 and 11)

```
SVC Optimization Results:
  Kernel   Gamma  Mean Accuracy  Standard Deviation
0    rbf   0.001       0.990837            0.003806
1    rbf   0.010       0.759146            0.058706
2    rbf   0.100       0.079889            0.010018
3    rbf   1.000       0.093659            0.033116
4    rbf  10.000       0.077266            0.009087
5   poly   0.001       0.986907            0.005066
6   poly   0.010       0.986907            0.005066
7   poly   0.100       0.986907            0.005066
8   poly   1.000       0.986907            0.005066
9   poly  10.000       0.986907            0.005066
```

Figure 4: Optimized Results for SVM Classifier

4 Discussion

After ensuring all the classifiers are optimized, it is clear that SVC and KNN achieved the highest mean accuracy , closely followed by Random Forest with around a difference of 0.1. This suggests that both SVC and KNN are well-suited for this classification task in comparison to Random Forest. However, although they performed better in terms of accuracy, Random Forest actually demonstrated superior performance in terms of standard deviation, indicating that while SVC and KNN might yield slightly better results, Random Forest would be able to produce results more consistently.

When evaluating the test set performance, SVC achieved the highest accuracy at 0.9926, followed by Random Forest at 0.9778, and KNN at 0.9704. These results align with the training performance shown in the graph, where both SVC and KNN demonstrated high accuracy but larger standard deviations. The Random Forest's lower standard deviation and solid test accuracy suggest it might be the most reliable choice if you need the results to be consistent throughout.

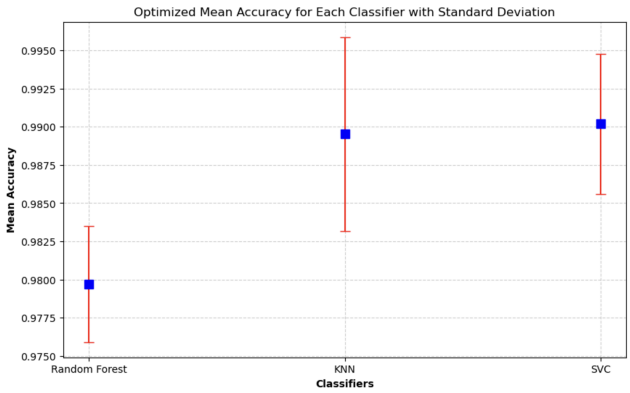

Figure 5: Final Comparison of Classifier Accuracy and Standard Deviation

5 Conclusion

The results from the confusion matrices confirm that Support Vector Machine is the most effective classifier for the MNIST dataset, achieving the highest accuracy meaning it performed the best on the unseen data with a low standard deviation. This makes SVM the optimal choice for this dataset. Where as KNN Performed nearly as well as SVM but showed less consistency. Random Forest while slightly less accurate compared to KNN and SVM, Random Forest exhibited the smallest standard deviation, showing its ability to reproduce values.

Overall, the findings indicate that SVM is the best-suited model for this classification task due to its balance of accuracy and stability, followed closely by KNN. Random Forest, while reliable, was outperformed by the other two classifiers.

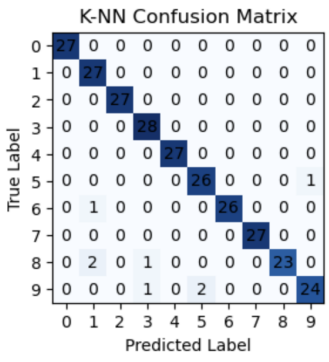

Figure 6: Confusion Matrix for Optimized K-NN

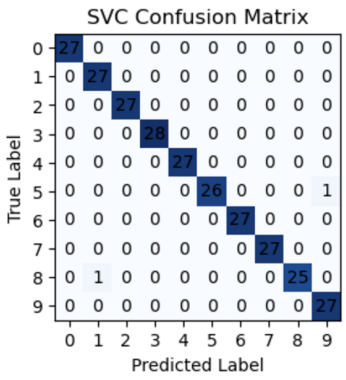

Figure 7: Confusion Matrix for Optimized SVC

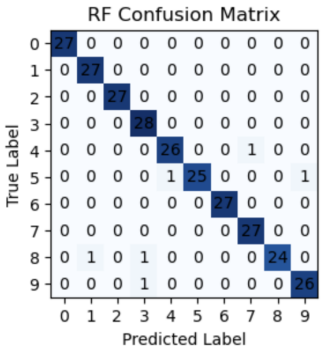

Figure 8: Confusion Matrix for Optimized Random Forest

6 Appendices

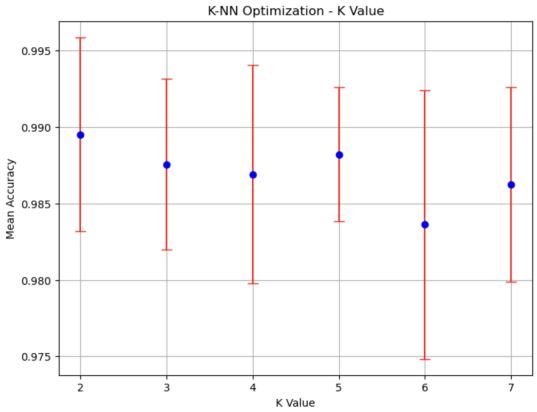

Figure 9: Hyperparameter Tuning Graph for K Nearest Neighbors

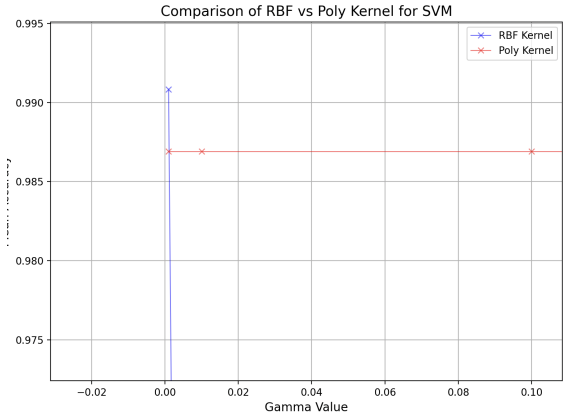

Figure 10: Hyperparameter Tuning Graph for SVC

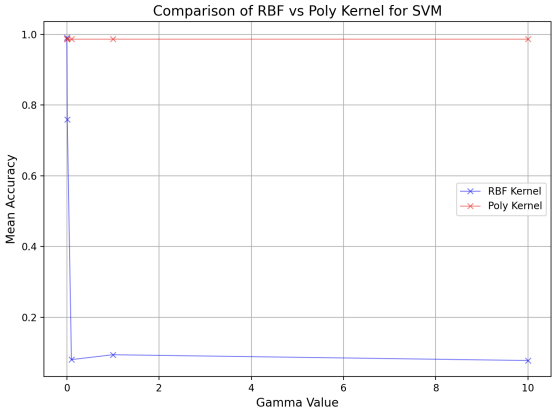

Figure 11: Hyperparameter Tuning Graph for SVC

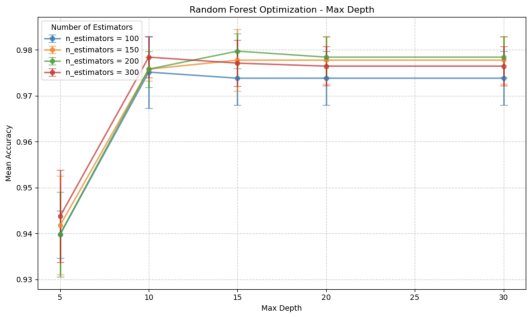

Figure 12: Hyperparameter Tuning Graph for Random Forest

7 References

References

[1] Scikit-learn developers, *KNeighborsClassifier*, 2024. *https://scikit-learn.org/dev/modules/generated/sklearn.neighbors.KNeighborsClassifier.html*

[2] Scikit-learn developers, *RandomForestClassifier*, 2024. *https://scikit-learn.org/dev/modules/generated/sklearn.ensemble.RandomForestClassifier.html*

[3] Scikit-learn developers, *SupportVectorClassifier*, 2024. Available at: *https://scikit-learn.org/dev/modules/generated/sklearn.svm.SVC.html*

[4] Steinbach, M. and Tan, P. N., *K-Nearest Neighbors*, In: P. N. Tan and M. Steinbach, *Data Mining: Concepts and Techniques*, Elsevier, 3rd edition, 2024.

[5] Testas, A. *Support Vector Machine Classification with Pandas, Scikit-Learn, and PySpark*, 2023,

[6] Hunter, J. D., *Matplotlib: A 2D Graphics Environment*, Computing in Science Engineering, Vol. 9, No. 3, 2007 Available at: https:// matplotlib.org/

[7] LeCun, Y., Cortes, C., and Burges, C. J. C., *The MNIST Database of Handwritten Digits*, 2010.