



Department of Statistical Sciences Plagiarism Declaration form

A copy of this form, completed and signed, to be attached to all coursework submissions to the Statistical Sciences Department. Submissions without this form will not be marked.

COURSE CODE: STA 5076Z

COURSE NAME: Supervised Learning

STUDENT NAME: Adam Mosam

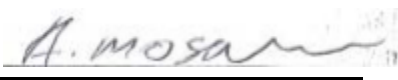
STUDENT NUMBER: MSMADA002

TUTORS NAME: NA TUT. GROUP #: NA

PLAGIARISM DECLARATION

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used a generally accepted citation and referencing style. Each contribution to, and quotation in, this tutorial/report/project from the work(s) of other people has been attributed, and has been cited and referenced.
3. This tutorial/report/project is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own work.
5. I acknowledge that copying someone else's assignment or essay, or part of it, is wrong, and declare that this is my own work.

Note that agreement to this statement does not exonerate you from the University's plagiarism rules (http://www.uct.ac.za/uct/policies/plagiarism_students.pdf).

Signature: 

Date: 02/06/2023

STA5076Z Supervised Learning Assignment 3

Adam Mosam (MSMADA002)

02 June 2023

Contents

1	Introduction	2
1.1	Dataset	2
2	Methodology	2
2.1	Data subset selection	3
3	Neural Network Model	4
3.1	Hyperparameter selection via cross-validation	4
3.2	Full neural network model	5
4	Support vector machines	5
4.1	Kernal selection	6
4.2	Hyperparameter selection via cross-validation	6
4.3	Full support vector machine model	7
5	Discussion & conclusions	7

1 Introduction

This report compares the performance of Support Vector Machines (SVM) and Neural Networks (NNs) in classifying handwritten digits as even or odd. The goal is to develop an accurate classification scheme for determining the parity of handwritten numbers. By examining implementation details, training processes and hyperparameter selection, this study identifies the strengths and weaknesses of each method. Insights gained from the performance analysis contribute to determining the suitability of SVM and NNs for this image recognition problem.

1.1 Dataset

The dataset used in this study consists of handwritten digits represented as greyscale images with a resolution of 28 by 28 pixels. Each pixel in the image has a greyscale intensity value ranging from 0 to 255, with 0 representing black and 255 representing white. These intensity values capture the tone of the pixel, providing information about the shading and texture within the digit image. By analyzing the variations and distributions of pixel intensities across the image grid, the classification algorithms can learn to distinguish the even and odd digits effectively. A visual representation of the image format is shown in Figure 1 for two observations from the dataset.

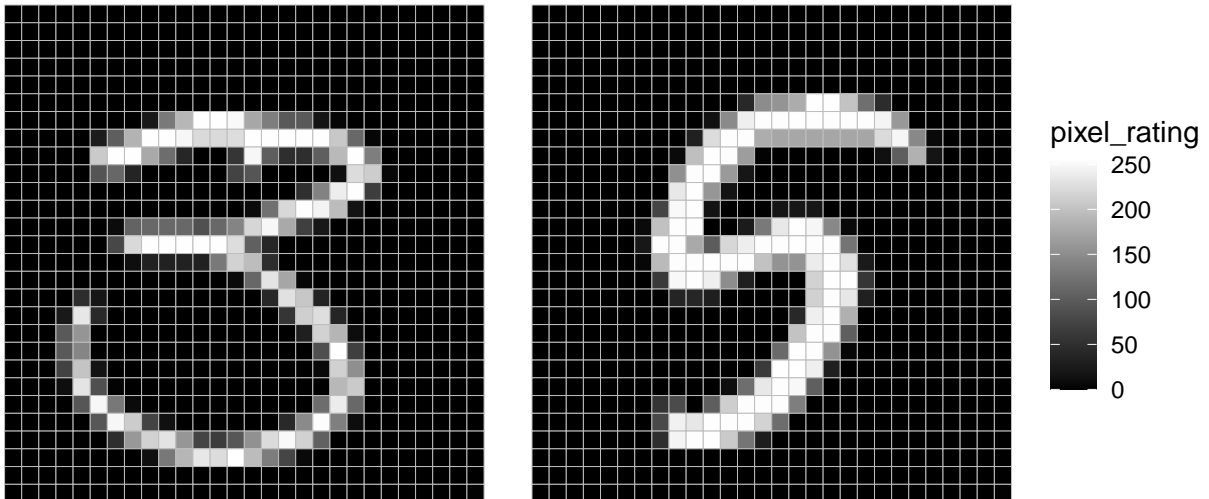


Figure 1: Image of a handwritten digit represented on a 28x28 pixel grid.

The first column of the provided training dataset, “evenodd”, represents the true label of each digit, indicating whether it is even or odd. The remaining columns, “pixle1” to “pixle784, represent the 28 by 28 grid.

2 Methodology

The process utilized for training the NN and SVM models in this investigation has been shaped by the constraints of limited computing power. Balancing model performance and simulation times, the following procedure has been proposed in Figure 2.

In Figure 2, a hybrid approach that combines the hold-out method and validation method is adopted. The validation method is applied to a subset of the main dataset (which consists of approximately 3,800

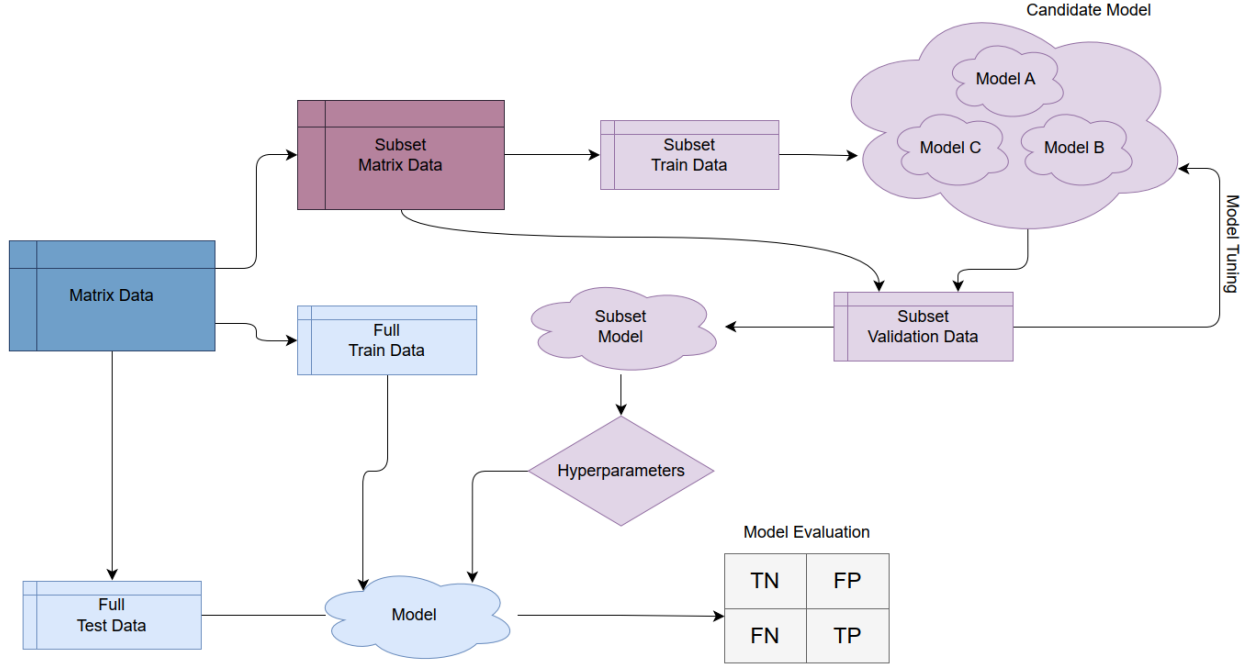


Figure 2: Preprocessing workflow

observations) and used for hyperparameter optimization. As will be discussed in Section 2.1, the reduced subset demonstrates high accuracy compared to models trained on the full dataset, suggesting that additional data does not significantly enhance performance. However, performing cross-validation with parameter optimization using grid search techniques consumes considerable computing power. Hence, the reduced subset will be used for parameter optimization, and the fine-tuned hyperparameters will subsequently be applied to the main training model, which encompasses the full dataset. The model will then finally be evaluated on the test set.

2.1 Data subset selection

To ensure consistency between the subset model and the full model, it is important to select a large enough subset for the training process and to achieve a reasonable level of accuracy using the subset data. In this regard, simple Neural Network (NN) and Support Vector Machine (SVM) models will be fitted against the different datasets using various subset percentages. The accuracy scores based on the test set will then be recorded.

The NN model, implemented using the h2o package, incorporates the following parameters:

- Hidden layers and neurons: A single layer with 200 neurons.
- L1 regularization: $\lambda = 1e-7$.
- Epochs: 10.

On the other hand, the SVM model, utilizing the Caret package, includes the following parameters:

- Kernel: Polynomial, with a degree of 2.
- Cost: 1.

As depicted in the figure below, a subset of 20% from the full dataset yields a remarkably high accuracy. Therefore, this 20% subset will be employed for the cross-validation hyperparameter tuning process.

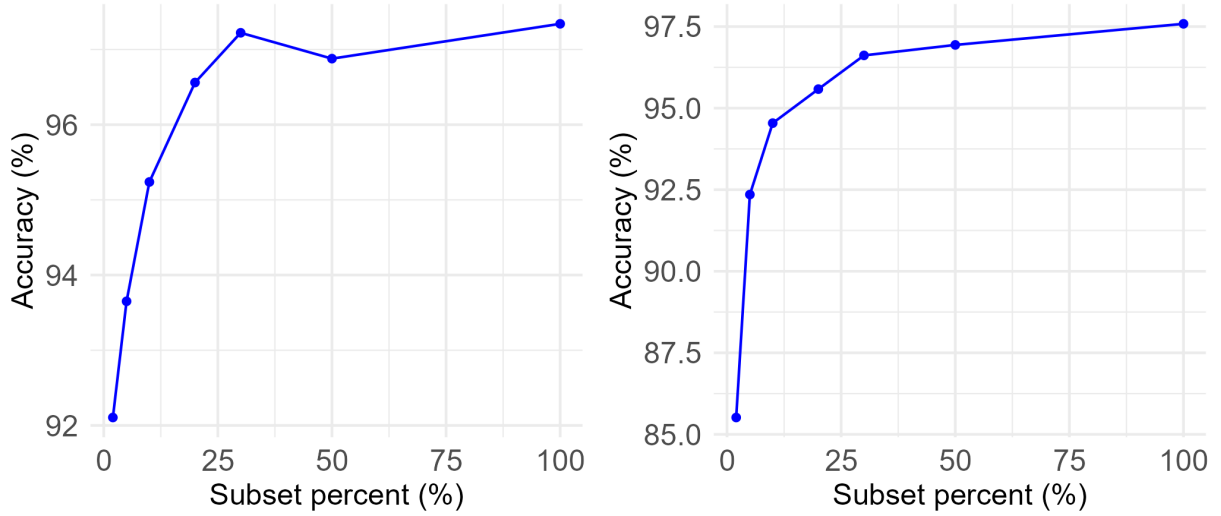


Figure 3: Accuracy vs subset percentage of dataset. (Right) Neural network model, (Left) support vector machine model

3 Neural Network Model

Neural networks (NN) is a supervised learning algorithm that aims at mimicking the functioning of the human brain. They consist of interconnected artificial neurons that process and learn from data to make predictions or classify information. Neural networks excel at handling complex and non-linear relationships in data, making them widely used in various domains.

In this investigation, we will utilize the h2o package to develop and train our neural network models. Among the available neural network packages, h2o stands out for its exceptional speed and efficiency. It leverages distributed computing capabilities, allowing for parallel processing and optimized memory management. As a result, h2o achieves faster training and prediction times, making it ideal for handling large-scale datasets and computationally intensive tasks.

3.1 Hyperparameter selection via cross-validation

To maximize model performance, an important step is to fine-tune the hyperparameters of the neural network model. In this study, a 20% subset of the dataset is utilized for cross-validation to select the optimal hyperparameters. 5 folds will be used. The cross-validation will involve splitting the subset into training and validation sets, training the model with different hyperparameter configurations, and evaluating the performance based on accuracy.

A grid search will be used to select the optimal hyperparameters, of which will include:

- Neurons: 1 layer with 50, 100 and 200 neurons, and 2 layers with 50, 100 and 200 neurons in each layer.
- L1 regularization: λ taken as 1e-3 and 1e-5.

Epoch will be taken as a fixed value of 10 (as it was found to produce good results). The figure below, highlights the variation in performance for the different hyperparameters.

As shown, the optimal model contains the following hyperparameters:

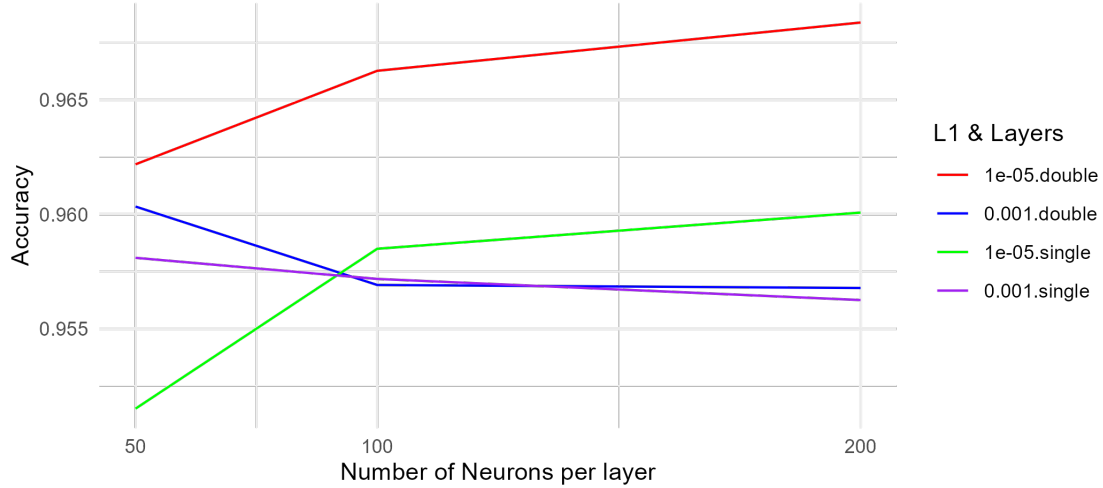


Figure 4: Cross-validation accuracy results, with hyperparameter search grid on L1 regularization penalty factor and neurons

- 2 layer with 200 neurons each.
- λ as $1e-5$

3.2 Full neural network model

The optimal hyperparameters, determined using a subset of the dataset, will be used in the main training model. The main training model employs the holdout method and is trained on the full training dataset. The model's performance on the test set is evaluated using metrics such as accuracy, recall (sensitivity), specificity, and precision. These metrics provide an assessment of the model's ability to correctly classify instances in the test set.

As shown in the table below, the model achieves a high accuracy, indicating excellent performance. This implies that the model has successfully classified a large proportion of instances correctly.

Table 1: NN results showing accuracy, specifisity, precision and recall

Accuracy	Precision	Recall	Specificity
0.9835	0.9841	0.9822	0.9848

4 Support vector machines

Support Vector Machines (SVM) are powerful machine learning algorithms widely used for classification and regression tasks. SVMs excel in handling complex datasets with non-linear relationships. They work by finding an optimal hyperplane that separates data points into different classes, maximizing the margin between the classes.

In this investigation, the caret package will be utilized to train and evaluate the SVM models. Caret provides fast and efficient implementations of SVMs, making it a suitable choice for our analysis. Compared to the e1071 package, Caret offers improved speed and efficiency, enabling faster training and prediction times.

The selection of kernal type and hyperparameter selection will be discussed in the sections top follow.

4.1 Kernel selection

The selection of the kernel type in Support Vector Machines (SVM) plays a vital role in achieving accurate classification. In this investigation, we will explore multiple kernel types, including linear, radial, polynomial, and sigmoid, to identify the one that best fits the underlying patterns in our data. The models will be run on 10% of the dataset, with an 80/20 split for training and test sets, respectively.

Based on the evaluation of different kernel types, the polynomial kernel stands out as the top performer among the models tested. It should also be noted that the other three kernel models took much longer to converge, resulting in lengthy computing times, compared to the polynomial model. This fast convergence of the polynomial model could suggest a compatible relationship with the data, which is highlighted in the test accuracy result. In contrast, the long convergence times of the other models may indicate that they are not suitable kernel types for the given data. Going forward, the polynomial kernel will be employed in the SVM models.

Table 2: Test accuracy using different SVM kernel functions

Linear	Radial	Polynomial	Sigmoid
0.7884	0.4683	0.9603	0.4683

4.2 Hyperparameter selection via cross-validation

The methodology for building SVM models will follow a systematic approach, similar to that for the NN model. Hyperparameter tuning will be performed on a subset of the dataset using cross-validation to identify the optimal set of hyperparameters. 5 folds will be used. These optimal hyperparameters will then be used to train the main SVM model on the full training dataset.

A grid search will be used to select the optimal hyperparameters, of which will include:

- Polynomial degree order: 2^{nd} , 3^{rd} and 4^{th} order polynomials will be tested.
- Cost: cost values of 0.01 and 10 be used.

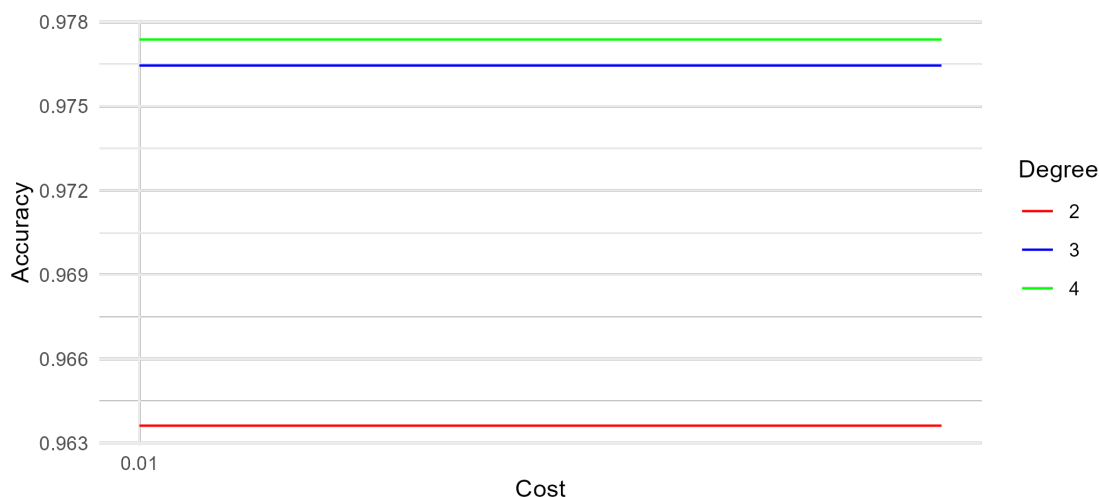


Figure 5: Cross-validation accuracy results, with hyperparameter search grid on cost and polynomial degree order

It may be observed that the cost remains at a constant accuracy for all degree orders. This could be due to the fact that the data fits well to the data using the polynomial kernel, as discussed in the last section. The degree order however, is shown to affect the accuracy score of the model, with an optimal degree of 4 observed.

4.3 Full support vector machine model

As was done in the previous section, the fined tuned hyperparameters will now be applied to the full NN model which will be trained on the entire training dataset. The model will then be evaluated on its performance in classifying the test set.

As shown in the table below, the model achieves a high test accuracy, indicating that model fitted well to the data. In addition, it should be noted that the accuracies are on par with those shown in the NN section.

Table 3: SVM results showing test accuracy, specifisity, precision and recall

Accuracy	Precision	Recall	Specificity
0.9798	0.9791	0.9819	0.9775

5 Discussion & conclusions

The results from the NN and SVM models are summarized below:

Table 4: NN and SVM results showing test accuracy, specifisity, precision and recall

Model	Accuracy	Precision	Recall	Specificity
NN	0.9835	0.9841	0.9822	0.9848
SVM	0.9798	0.9791	0.9819	0.9775

The results show that both the neural network (NN) and support vector machine (SVM) models perform similarly, with the NN model achieving a slightly higher accuracy score. The accuracy score for both models is notably high at 98%.

Based on these findings, it can be concluded that both models effectively predict the outcome of the data with a high level of accuracy. However, the NN model outperforms the SVM model, making it the preferred choice in this scenario.