1.1: Introduction

This research paper uses Artificial Neural Networks (NN) and Support Vector Machines (SVM) to classify the blocks of a page layout of a document that have been detected by a segmentation process. There are five classes in total, namely, text, horizontal line, picture, vertical line and graphic. These classes are associated with levels 1 to 5, respectively, in the dataset. The objective of this research is to construct NN and SVM models that are able to correctly classify each category of class with the least misclassification error rate. Thus either a SVM or NN model must be able to determine class for a given attribute combinations. The classification output of these models will be compared to determine which model had better classification properties. In particular, model performance would be evaluated based on misclassification error rate on unseen data.

The paper will commence by providing an overview of the data and methodology employed in the model building process. This will include a short description of both the NN and SVM modelling techniques. Subsequently, each model output would be provided along with some critical statistics such as confusion matrices and classification metrics.

1.2: Data and Methodology

1.2.1: Data

The data used in this paper was provided by the Department of Statistical Sciences of the University of Cape Town. The training data consists of 4925 observations which are taken from 54 different documents. There are 11 associated variables of which class is one. The description of each variable as per assignment document is as follows:

• height: Height of the block.

• length: Length of the block.

• area: Area of the block (height × length).

• eccen: Eccentricity of the block (length / height).

• p black: Percentage of black pixels within the block (blackpix / area).

• p and: Percentage of black pixels after the application of the Run Length Smoothing Algorithm (RLSA) (blackand / area).

• mean tr: Mean number of white-black transitions (blackpix / wb trans).

• blackpix: Total number of black pixels in the original bitmap of the block.

• blackand: Total number of black pixels in the bitmap of the block after the RLSA.

• wb trans: Number of white-black transitions in the original bitmap of the block.

A separate test data consisting of 548 observations is also provided. The ‘class’ variable is excluded from the test data, and the objective of the research is to use the resulting best classification model from NN and SVM to estimate ‘class’ given all other attributes of the test data.

1.2.2: Methodology

1.2.2.1: Data Normalization

Both the NN and SVM approach require the that data be measured on the same scale in order to not give larger observations within each attribute too much importance, as this might skew the model output. To this end, a min-max normalization approach is used to normalize the data. Min-Max normalization maps each observation to a range between 0 and 1. Thus the largest observation of a particular attribute gets transformed to 1, while the least observation retains a value of 0. For a given observation, x_i , associated normalized value, z_i , is given by the following equation:

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

After the given training data has been normalized, it is randomly split into training and test data set using an 80/20 split. Thus 80% of the training data is used to train the models, while the remaining 20% is reserved for testing model performance.

1.3: Description of each technique

1.3.1: Support Vector Machines

Support Vector Machines refers to a classification method whose objective is to find an N-dimensional hyperplane that can distinctly classify data into different categories. This objective is achieved by finding a hyperplane that maximises the distance between the hyperplane and boundary lines. The datapoints on each side of the hyperplane theoretically belongs to a different class.

The general hyperplane equation can be expressed mathematically as follows:

$$y = w^T x + b$$

Where y is the classification label; and w and b are parameters of the planes

1.3.2: Artificial Neural Networks

Artificial Neural Networks are biologically inspired algorithms that model the relationship between some given input signals and output signals. They are able to recognise and interpret sensory data such as numerical data, images, speech and text. Neural networks are mainly a classification method, and are best suited for applications with well-defined input and output features. For classification procedures, the neural network can be expressed an optimization of the following cost functional:

$$J(w) = \frac{1}{n}[\sum_{i=1}^{n}\sum_{k=1}^{K} y_k^{(i)} \log\left(f_w\left(x^{(i)}\right)k\right) + \left(1 - y_k^{(i)}\right)\log\left(1 - f_w\left(x^{(i)}\right)k\right] \quad + \frac{\lambda}{2n}\sum_{l=1}^{L-1}\sum_{i=1}^{sl}\sum_{j=1}^{sl+1}(w_{ji}^{l})^2$$

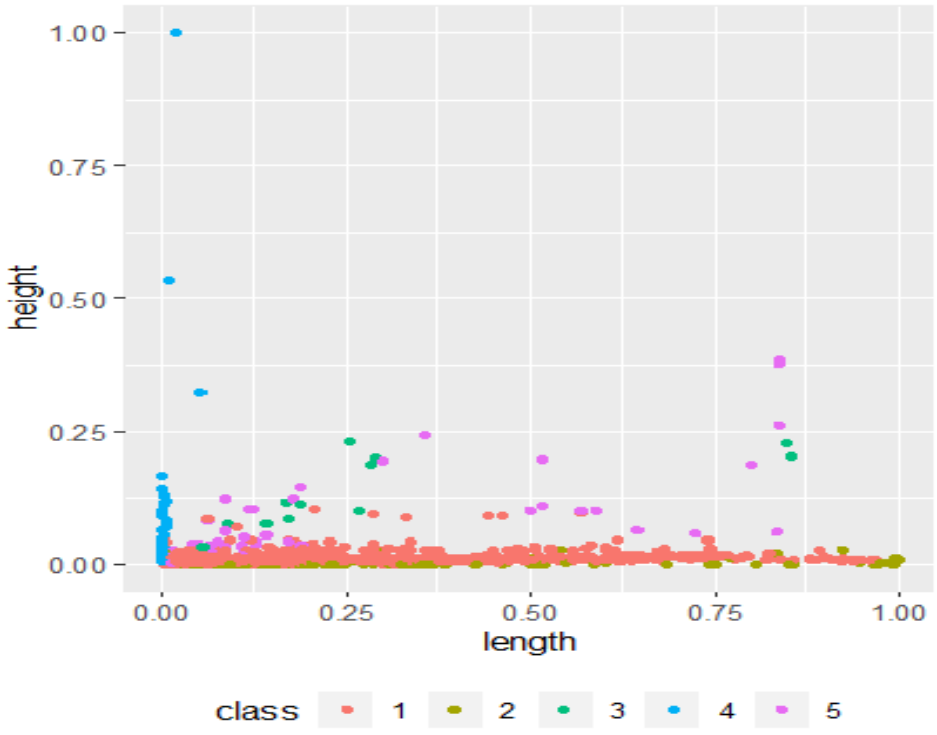
Where the objective is to minimize, $J(w)$, using Gradient Descent Algorithm.

2: Model Results

2.1: Support Vector Machines

We begin by producing a ggplot of the training data on different attribute combination to get an idea of how the data is separated. Plot 1.1 below displays height and length attributes colour coded based on class variable. From the plot it is evident that the classes are not linearly separated but rather are distributed randomly on the plot. Other attribute plot combinations (not reported here) showed the same pattern. This immediately implies that a nonlinear hyperplane is a suitable classification method to apply for this dataset.

Plot 1.1: Distribution of Class on Height and Length



Initial SVM models were produced using the radial and the polynomial kernels respectively, and a cost function of 5. Of the two types of kernels used, the radial kernel model had low misclassification error rate, about 7.4% versus 9.8% for the polynomial kernel. This suggests that the radial kernel has better classification properties relative to the polynomial kernel. As a result, the radial kernel model is the base model upon which we train our data. The summary of the radial kernel model is displayed in appendix, SVM Summary1. This summary shows that the model had 618 support vectors in total, of which 268 belong to class level 1. Holding all the other parameters constant at 0.25, the classification plot of the training SVM model on length and height is displayed in plot 1.2 below. The plot shows the data, decision boundaries and support vectors which are represented by the crosses. It is evident from the plot that the dataset is nonlinear. For instance, there seem to be an overlap in the datapoints and support

vectors across classes. Furthermore, some predicted classes, such as, the second class seem to contain no data.

Plot1.2: SVM Classification Plot

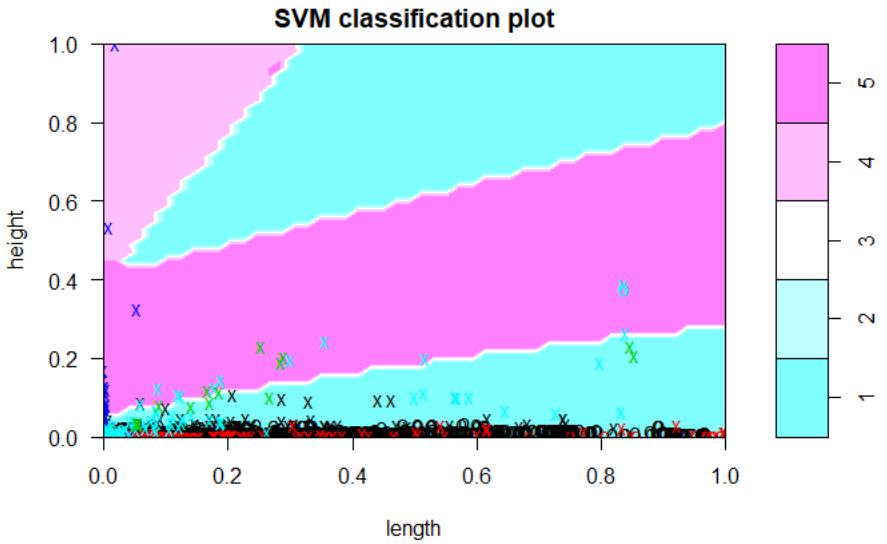


Table1.1: Training SVM Confusion Matrix

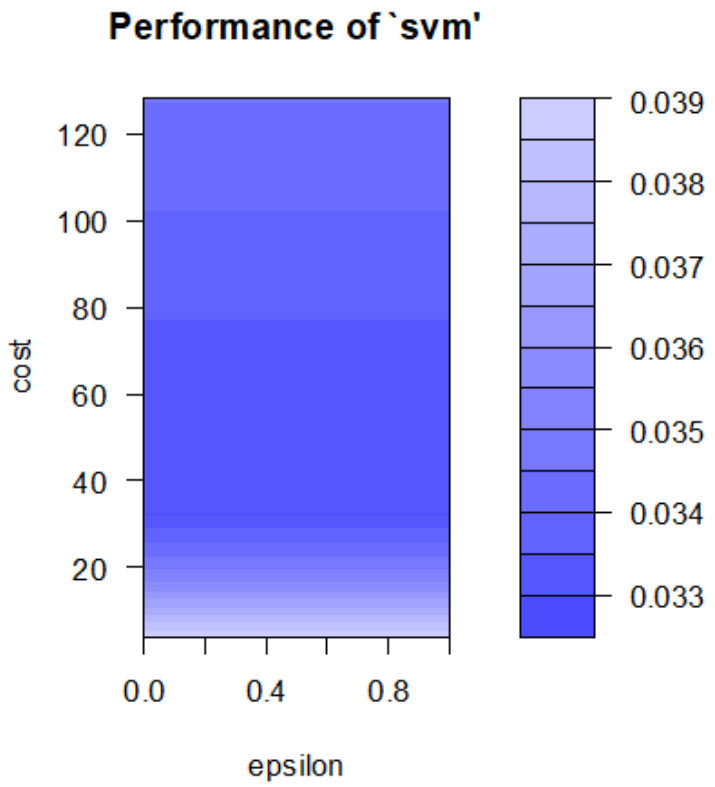
		Actual				
		1	2	3	4	5
	1	3527	132	14	63	78
	2	5	107	0	1	0
Predicted	3	0	0	4	0	0
	4	0	0	0	3	0
	5	0	0	0	0	6

The initial SVM model resulted in the confusion matrix illustrated in table1.1 above. The confusion matrix shows that 3647 training model data values are correctly classified and the remaining 293 values are misclassified. This accounts for approximately 93% correct classification rate and a 7% misclassification error rate.

To obtain better classification, we tuned the SVM model parameters using the tune function. The objective was the test several different in order to identify the variables that minimizes the misclassification error rate in a 10-fold cross validation. To this end, the epsilon and cost parameters were tuned. In particular, epsilon was set as a sequence of values from 0 to 1 with 0.1 increments in each sequence. The cost parameter was set to 2 raised to power of an arbitrary range of 2 to 7. The highest power was capped at 7 to avoid having a high constraint violation cost at this might result in a high penalty for non-separable points, which might lead to model overfitting. The plot of the tuning model is displayed in plot1.3 below and it's summary is presented in the appendix, Tune Summary2. Tuning suggests that the best model is one with the parameters of 0 and 32 for epsilon and cost respectively. Thus from tuning we derive our best model and this model is the *de factor* model for testing unseen data. The summary of the best model is also presented in the appendix, SVM Summary 3. It can be observed from the

best model summary that the number of support vectors have decreased to 373, with now 166 support vectors associated with level 1 class. The resulting classification plot is illustrated in plot1.4 in the appendix. The plot indicates that class level 1 is the most pronounced.

Plot1.3: Tuning Parameters



To evaluate the improvement in classification performance from the initial training model to the best model obtained after tuning the model parameters, another confusion matrix was constructed and the associated misclassification error rate calculated. The best SVM model's performance was also tested on the unseen training data. The confusion matrices obtained from training data and unseen training data are tabulated in table 1.2 and table 1.3 below. In best SVM confusion matrix, we can see that only 81 values are misclassified. This represents a significant improvement in model classification from the initial. In total misclassification has decreased from about 7% in the initial model to a mere 2% misclassification error rate on the best model. This implies that tuning the parameters to obtain the best model had a pronounced effect on our SVM model. The effect of tuning is further illustrated by the model's performance on unseen training data. In the unseen data, the model correctly classified all values except for 31. This represents misclassification error rate of approximately 3%. Thus the model's performance on the unseen appears to be better than that of the initial model based on the misclassification error rate. The misclassification error rates of the best SVM model and unseen training data are tabulated in table 1.4 in the appendix.

Table1.2: Best SVM Confusion Matrix

		Actual				
		1	2	3	4	5
	1	3515	20	0	9	30
	2	13	216	0	1	0
Predicted	3	2	0	18	0	0
	4	2	3	0	57	0
	5	2	0	0	0	54

Table1.3: Unseen Training Data Confusion Matrix

		Actual				
		1	2	3	4	5
	1	880	8	3	2	9
	2	6	56	0	0	0
Predicted	3	2	0	2	0	0
	4	0	0	0	9	0
	5	0	0	0	1	7

The final leg of the SVM section is to predict class using the given unlabelled dataset. The results of this prediction are presented on the spreadsheet that is submitted with this research report. The spreadsheet is titled SVM_class_Prediction and contain ID and class attributes. However, the predicted class values from this data with no labels is illustrated in table 1.5 below. The table shows that about 480 values were predicted to be in class level 1, 38 in level 3, 5 in level 3 and 25 in level 5. No value was predicted to be in class level 4.

Table1.5: No Labels Test Data Predicted Class Values

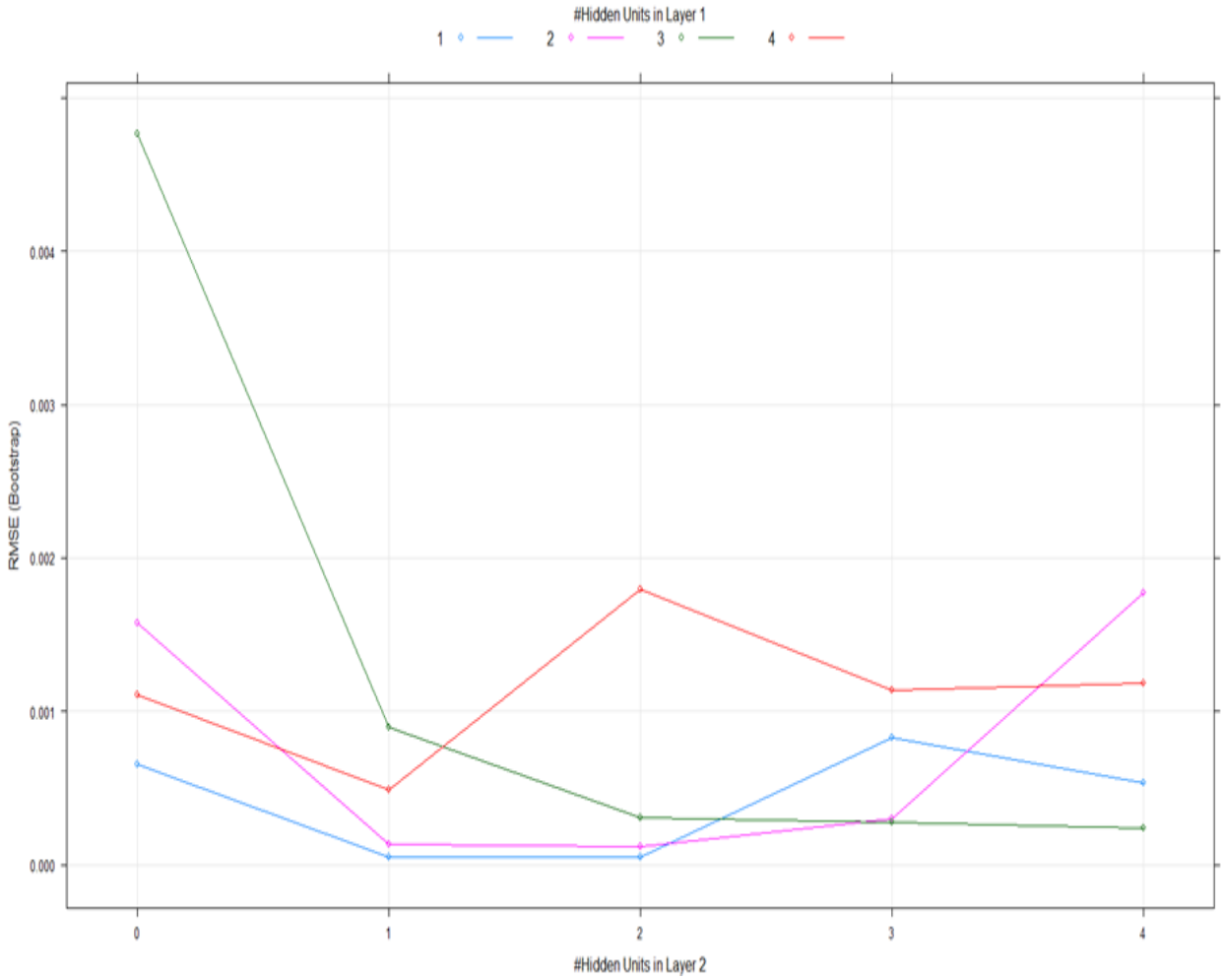
NoLabels Data					
Class	1	2	3	4	5
Predicted	480	38	5	0	25

2.2: Artificial Neural Networks

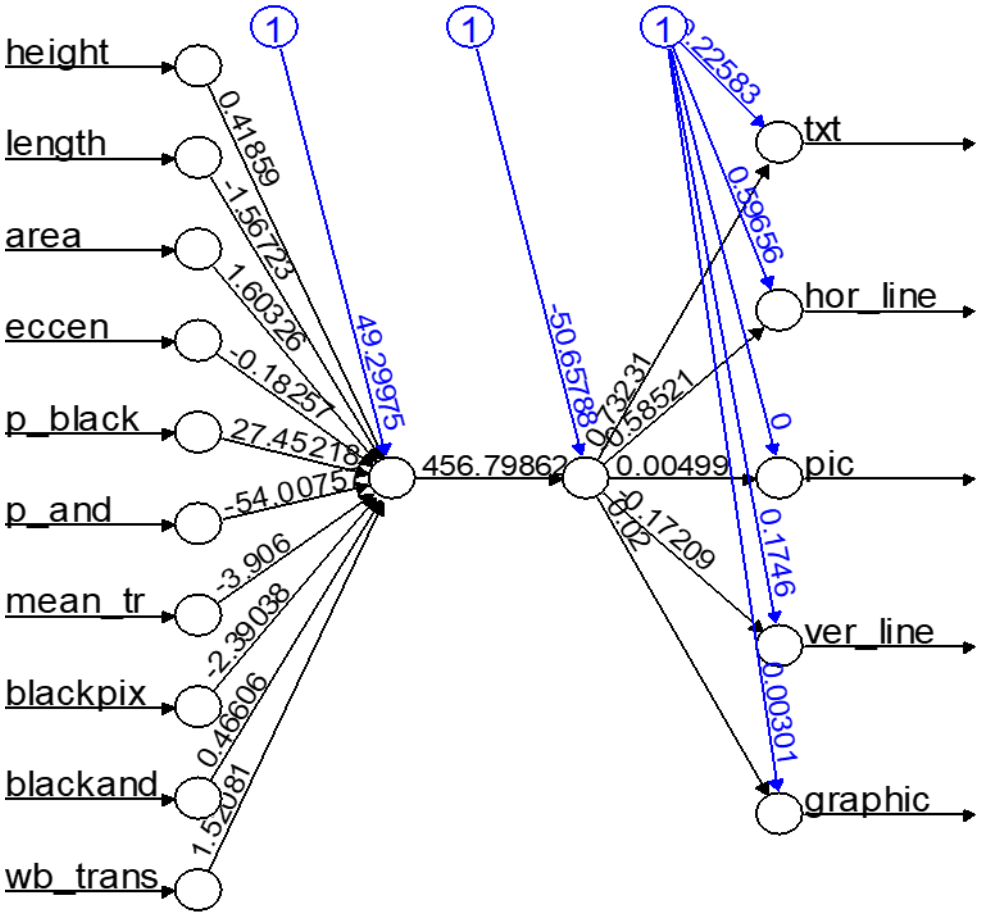
The neuralnet() function is used to build an artificial neural network (nn) classification model. Since this is a multi-class classification problem we first had to encode categorical variables for the class attribute. The class attribute had 5 labels, therefore we replaced these with 5 variables namely; txt, hor_line, pic, ver_line, and graphic. This resulted in a dataset with 15 attributes. The predictor variables are all numeric so there was no need to encode them.

The data was trained using the train() function and neuralnet training method at a learning rate of 0.01. The training model plot is displayed below in plot 2.1. Using the train() function to tune the model parameters we can determine which parameters are the best fit for our nn model. In plot2.1 it is evident that the hidden layer parameters that result in minimum root mean squared error (RMSE) is a one:one vector hidden layer model. Therefore, an optimal model is built using these parameters. The resulting best model plot is displayed in plot2.2 below.

Plot 2.1: RMSE Against Hidden Layers



Plot2.2: Best Model Plot



Measuring Model Performance

Table2.1: NN Train Confusion Matrix

		Actual				
		graphic	hor_line	pic	txt	ver_line
Predicted	hor_line	1	198	0	75	58
	txt	83	41	18	3457	9

Table2.1 displays the confusion matrix from our best nn model. The table shows that the model only categorised the data into two classes; hor_line and txt. In its classification the model correctly predicted 3655 values. This represents approximately 7.2% misclassification error rate.

The model performance was also computed on the unseen training dataset. The confusion matrix from the training test data is presented in table2.2. Similar to table2.1, only two classes are also predicted for this dataset as well. The table shows that 924 values are correctly classified. This implies a misclassification error rate of about 6.2%. This is a rather striking

result as it shows that the misclassification error rate on unseen training data has improved compared to the misclassification observed on the training data. This possibly suggests that the model is not overfitting the data and thus is capable of retaining good predictions on different datasets that have not been seen before.

Table2.2: Unseen Training Data Confusion Matrix

		Actual				
		graphic	hor_line	pic	txt	ver_line
Predicted	hor_line	0	56	0	20	8
	txt	16	8	5	868	4

Finally, the nn model was also used to predict class using the given unlabelled dataset. The results of which are presented in a spreadsheet submitted along with this research report. The spreadsheet is titled NN_Prediction and contain ID and class attributes.

3: Comparing the Models

The SVM and NN models presented above can be compared on two levels. The first comparison can be made on the practical nature of the model – for this, we will use the results presented above as a reference factor. The second comparison can be made based on the models' theoretical appeal.

From a practical standpoint, the above results suggest that the SVM model has far better predictive properties than the NN model. For instance, the SVM model was able to categorize class for all class levels. On the other hand, the NN model only classified the values to only two categories. Moreover, NN model had a higher misclassification error rate than SVM model for both the training data and unseen training data. However, we also saw that the NN model experienced a reduction in its misclassification error rate which suggests that the model might not be overfitting the data and thus might be capable of an overall good performance across on various unseen datasets. Nonetheless this conjecture is an empirical question.

Theoretically, the NN model has the advantage that it is robust with noisy data. However, it suffers from an interpretability disadvantage. The NN model is also computationally intensive and requires longer training time. On the other hand, SVM models have the advantage of being effective in high dimensional spaces, they are memory efficient – which implies that less computational resource requirement. Further, SVM models are versatile because they allow for specification of different kernel functions for different decision requirements. Custom kernels can also be specified. However, its major disadvantages are it performs poorly when the number of feature parameters are greater than the sample size. SVMs also cannot directly provide probability estimates.

4: Conclusion

In this research paper we used the Support Vector Machines and Artificial Neural Networks classification procedures in order to determine class label for a given feature combinations. The dataset had 10 explanatory features and 4925 observations. The model's performance was evaluated using the misclassification error rate derived from the confusion matrix. Overall the findings of the paper showed that the Support Vector Machines approach had better classification properties than the Neural Networks approach. This was evidenced by the SVM approach having a low misclassification error rate on both the training dataset and unseen training dataset.

References

Hastie, T., Tibshirani, R., & Freidman , J. (2008). *The Elements of Statistical Learning* . New York : Springer .

James , G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning: With R Applications.* New York: Springer Science + Business Media.

Machines, S. V. (2019, May 02). *Support Vector Machines*. Retrieved from Support Vector Machines: http://www.svms.org/

Appendix

SVM Summary1

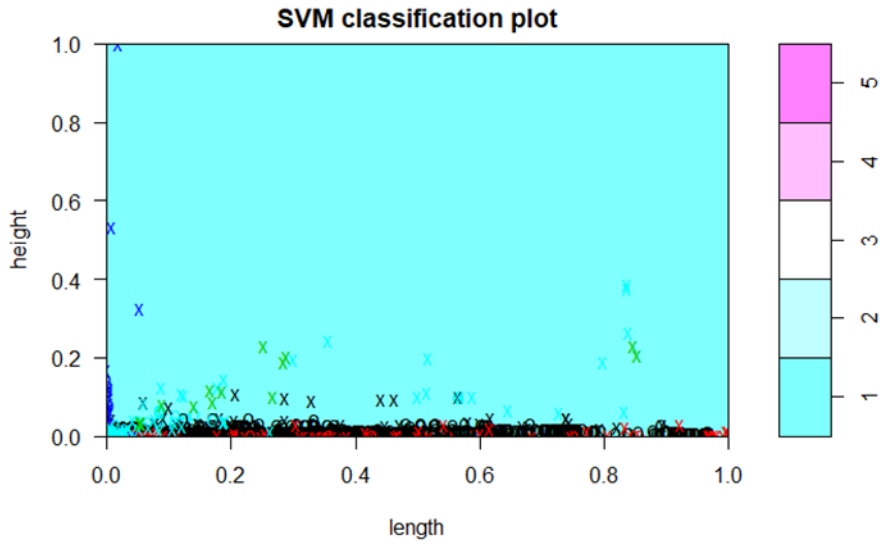
```
summary(svm_model)
Call:
svm(formula = class ~ ., data = train_data_svm, kernel = "radial", cost = 5
, scale = FALSE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  5
      gamma:  0.1

Number of Support Vectors:  618

 ( 268 182 67 83 18 )


Number of Classes:  5

Levels:
 1 2 3 4 5
```

Tune Summary2

```
> summary(tune_model)

Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation

- best parameters:
 epsilon cost
       0   32

- best performance: 0.03299492
```

SVM Summary3

```
> summary(best_svm_model)

Call:
best.tune(method = svm, train.x = class ~ ., data = train_data_svm, ranges
= list(epsilon = seq(0, 1,
    0.1), cost = 2^(2:7)))


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  32
      gamma:  0.1

Number of Support Vectors:  373

 ( 166 93 28 68 18 )


Number of Classes:  5

Levels:
 1 2 3 4 5
```



Plot1.4: Best SVM Classification Plot

Table1.4: Misclassification Error Rates

Misclassification Error Rate	
Best Model	Unseen Training Data
0,0208	0,0315

R Code:

See separate attached PDF