Chapter 6

Classification of Diabetes Mellitus Using Neural Network and Support Vector Machine

6.1 Introduction

Research in the field of neuroscience and artificial intelligence at present, is of vital importance because of the large number of patients with diseases such as diabetes and diabetic neuropathy [82]. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped in to that same space and predicted to belong to a category based on which side of the gap they fall on in addition to performing linear classification[53].

SVM that classifies novel points on the nodes and only transmits key information such as support vectors across the network. The generalization capability comes from the fact that SVMs are large margin classifiers. This means the training algorithm seeks to separate the positive points from the negative points, if the learning process is a binary classification problem. This is achieved by trying to separate the two classes of points with a plane that is equidistant from the closest points of each class. If there were two measurements on a line and each one belonged to different classes, then the SVM would place the discriminate function mid-way between the two points. Some classification problems are either so noisy or impossible to separate completely due to the lack of information in the chosen dimensions describing properties of the class instances. Noise is present when poor measurements are part of the problem domain or there are lacking key attributed to achieve separation of the data points. Misclassification is related to those data instances that cannot be classified correctly during the training process of the model.

6.2 Literature Survey

Sl No	Methodology	Authors, Year	Description
1	SVM-Fuzzy	Thirumalaimuthu Thirumalaiappan Ramanathan, Dharmendra Sharma,2015 [67]	Determines the efficiency in classifier by optimizing selection of right sized datasets. The levels of risks from data is classified by fuzzy reasoning. SVM is used to design rules in fuzzy system by using the SVM classification in rule inference.
2	Least Square Support Vector Machine (LS-SVM) with Particle Swarm Optimization (PSO)	Omar S. Soliman, Eman AboElhand,2014 [70]	LS-SVM algorithm is used for classification by finding optimal hyper plane which separates various classes. Modified PSO algorithm is used as an optimization technique for LS-SVM parameters to provide robustness of hybrid algorithm by searching for optimal values of LS-SVM parameters.
3	Linear Genetic Programming	K. Menaka, S. Karpagavalli, 2013 [68]	Linear GP was used and was run using Discipulus.10 –fold cross validation was performed for predicting the accuracy.
4	Multi-Layer Neural Network(MLNN) & Probabilistic Neural Network(PNN)	Hasan Temurtas , Nejat Yumusak, Feyzullah Temurtas , (2009) [77]	MLNN was trained using Levenberg-Marquardt (LM) algorithm. PNN with single hidden layer of locally tuned units fully interconnected to output layer was considered. Real valued input vectors were taken as feature vectors.10- fold cross validation was performed to compare the accuracy of neural networks.

6.3 Artificial Neural Network using Scaled Conjugate Gradient Back propagation Algorithm

The basic back propagation algorithm adjusts the weights in the steepest descent direction i.e. in the most negative gradient, in the direction in which the performance function is decreasing most rapidly. It turn south at although, the function decreases most rapidly about the negative gradient, this doesn't necessarily produce the fastest convergence .In the Conjugate Gradient (CG) algorithm a search is performed along a direction which produces fastest convergence than the steepest descent direction [72] , while preserving the error minimization achieved in all previous steps. This direction is called as conjugate direction. In most of the CG algorithm the step size is adjusted at each iteration. A search is made along the CG direction to determine the step size, which will minimize the performance function along that line. The entire CG algorithm starts out by searching in the steepest descent direction at first iteration. Frequently the CG algorithms are used with line search. The step size is approximated with line search technique, avoiding the calculation of Hessian matrix to determine the optimal distance to move along the current search direction. The next search direction is determined so that it is conjugate to previous search direction [72] .

Scaled CG algorithm doesn't requires line search at each iteration like other conjugate training functions. Step size scaling mechanism is used which avoids time consuming line search for learning iteration. Hence this algorithm makes it faster than any other second order algorithm [72]. The TRAINCG function requires more iteration to converge than other CG algorithms, but the number of computations in each iteration is significantly reduced because no line search is performed.

6.4 Support Vector Machine[70]

The foundations of Support Vector Machine (SVM) have been developed by Vapnik (1995) and has gained popularity due to many attractive features and promising empirical performance [66]. They are used for learning to predict future data. SVM are set of related supervised learning methods for classification and regression. SVM is a classification and regression prediction tool that uses machine learning approach to maximize predictive accuracy while automatically avoiding over fitting of data. SVM simultaneously minimizes the empirical classification error and maximizes the geometric margin. So, SVM is called maximum margin classifiers. It is a general algorithm based on guarantee risk bounds of statistical learning theory, Structural Risk Maximization (SRM) principle [71]. SVM can efficiently perform nonlinear classification. Kernel trick is used to construct mapping into high dimensional feature space. The computation is critically dependent upon the number of training patterns and to provide a good data distribution for a high dimensional problem will generally require a large training set.

The basic idea behind the SVM technique is to construct an n-1 dimensional separating hyper plane to discriminate two classes in a n dimensional space [68]. A data point is viewed as n dimensional vector. When two variables in a dataset are involved, it will create a two dimensional space, the separating hyper plane would be a straight line (one dimensional) dividing the space in half. Fig. 6.1 shows the classification plot of Diabetes data.

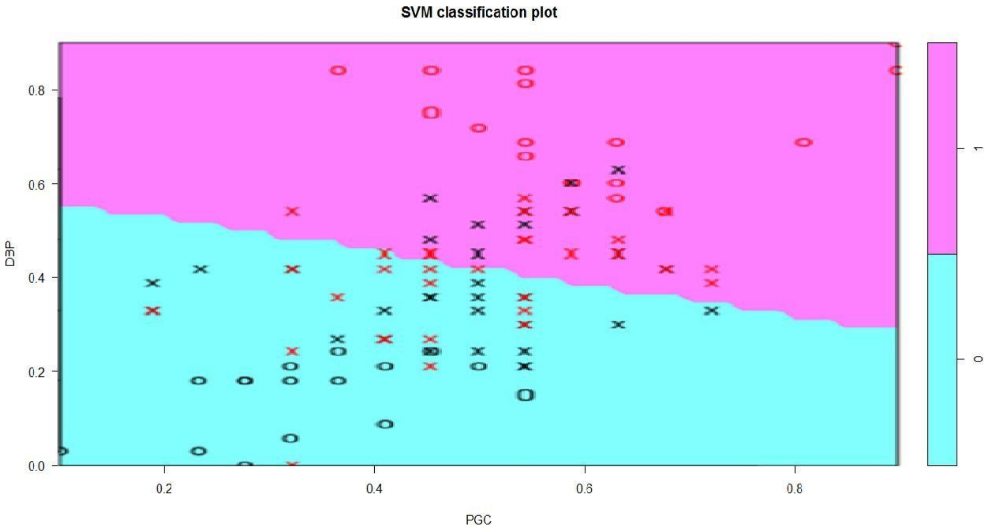


Fig.6.1. Classification plot of Diabetes data

When more dimensions are involved SVM searches for an optimal separating hyper plane called the maximum margin separating the hyper plane. The distance between the hyper plane and the nearest data point on each side called support vectors is maximized [66]. However some data points in the two classes might fall into an area that is not easy to be separated. SVM solves this problem by allowing some data points to the wrong side of the hyper plane. By introducing a user specified parameter C that specifies trade-off between the minimization of the misclassification and maximization of margin.

Given labeled training data as data points of the form M= {(x1, y1), (x2, y2)…., (xn,yn)} where yn= -1 or +1, a constant that denotes the class to which the data point xn belongs. N is the number of data sample. Each xn is a p-dimensional real vector. The SVM classifier maps the input vectors into decision value, and then performs the classification using an approximate threshold value [68]. A hyper plane can be represented by set of points x satisfying

$$w.\ x-b=0 \qquad (7.1)$$

w is normal vector to the hyper plane. The parameter b/|w| determines the offset of the hyper plane from the origin along the normal vector w, b is a scalar. The diagrammatic representation of hyper plane of SVM trained with sample soft w o classes is shown in Fig.6.2. Thedistancebetweenthehyperplaneis2/||w||.To minimize |w|, we need to ensure that for all I either

$$w.x-b= 1 \qquad \text{and} \qquad (6.2)$$

$$w.x-b= -1 \qquad (6.3)$$

$$w.x_i-b \geq 1\ ,\ \text{if } y_i=1 \qquad \text{or} \qquad (6.4)$$

$$w.x_i-b \leq 1\ ,\ \text{if } y_i= -1 \qquad (6.5)$$

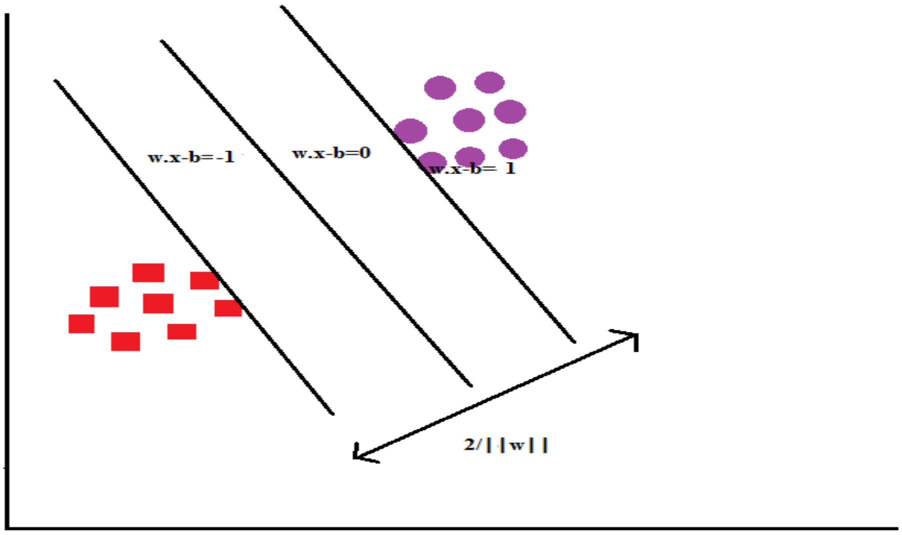
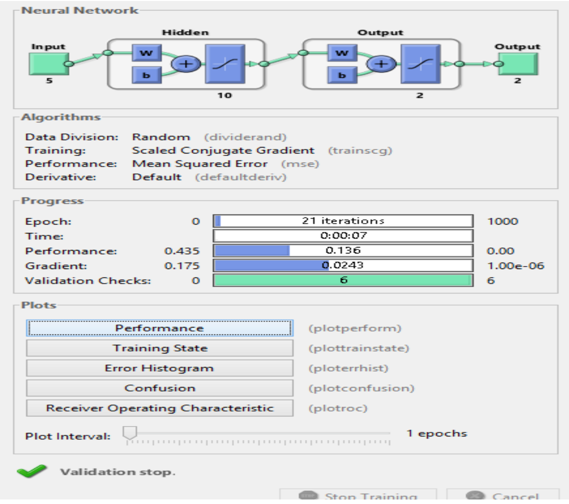


Fig 6.2. SVM hyper plane

6.5 Results of Artificial Neural Network using Scaled Conjugate Gradient Back Propagation Algorithm

Neural Network toolbox from MATLAB 13 was used to evaluate the performance of the neural network using TRAINSEG training function. The problem of training an ANN can be formulated as the unconstrained minimization problem of the error function E(w).CG methods are well suited for large scale neural networks due to their simplicity and their low memory requirement. The main objective is to test the training and generalization performance of the algorithm using diabetes dataset. 75% of the data was used for training, 15% of the data was used for validation and 10% of the data was used for testing.Fig.3.3 shows the MSE is equal to 0.136. Fig 6.4 shows the best validation performance is 0.292 at epoch 15.



Algorithms
Data Division: Random (dividerand)
Training: Scaled Conjugate Gradient (trainscg)
Performance: Mean Squared Error (mse)
Derivative: Default (defaultderiv)

Progress
Epoch: 0 | 21 iterations | 1000
Time: 0:00:07
Performance: 0.435 | 0.136 | 0.00
Gradient: 0.175 | 0.0243 | 1.00e-06
Validation Checks: 0 | 6 | 6

(plotperform)
(plottrainstate)
(ploterrhist)
(plotconfusion)
(plotroc)

Plot Interval: 1 epochs

✓ Validation stop.

Fig. 6.3. Neural Network performance analysis

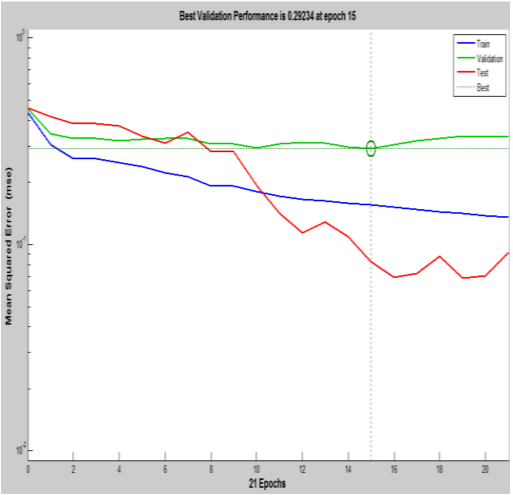
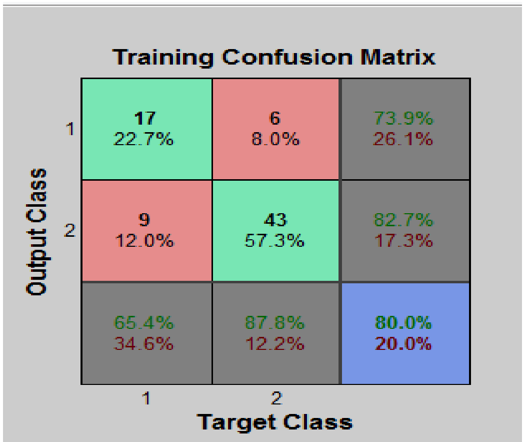


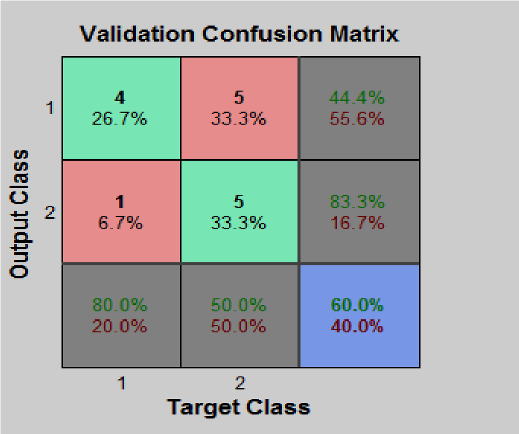
Fig. 6.4. Neural Network Training

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. Fig. 6.5 shows the confusion matrix for the training data. Fig 7.6 shows the confusion matrix for valid action data and Fig.7.7shows the confusion matrix for test data. The overall confusion matrix is shown in Fig 6.8.



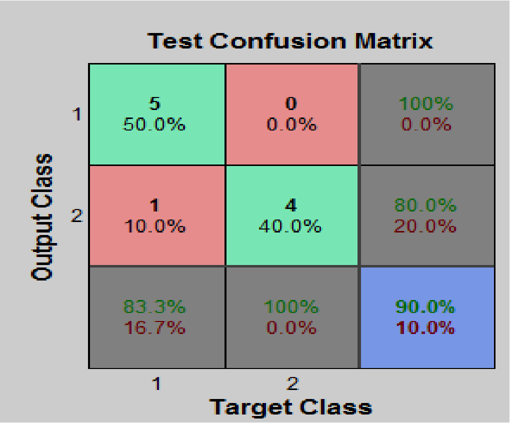
	1	2	
1	17 22.7%	6 8.0%	73.9% 26.1%
2	9 12.0%	43 57.3%	82.7% 17.3%
	65.4% 34.6%	87.8% 12.2%	80.0% 20.0%

Fig 6.5. Confusion matrix for training data



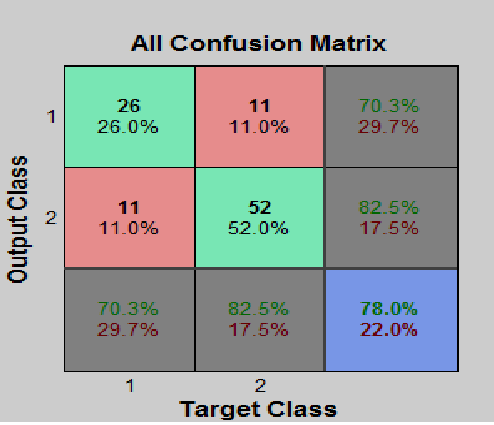
	1	2	
1	4 26.7%	5 33.3%	44.4% 55.6%
2	1 6.7%	5 33.3%	83.3% 16.7%
	80.0% 20.0%	50.0% 50.0%	60.0% 40.0%

Fig6.6 Confusion matrix for validation data



	1	2	
1	5 50.0%	0 0.0%	100% 0.0%
2	1 10.0%	4 40.0%	80.0% 20.0%
	83.3% 16.7%	100% 0.0%	90.0% 10.0%

Fig 6.7.Confusion matrix for test data



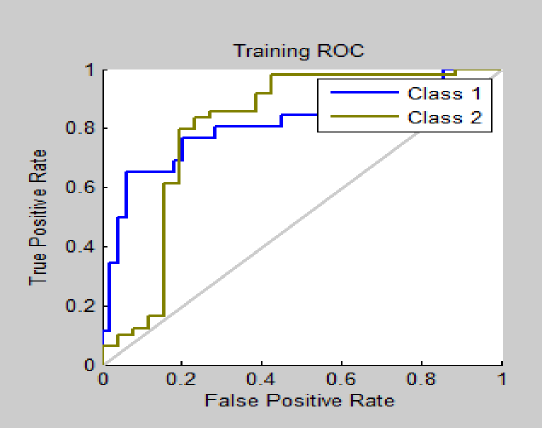
	1	2	
1	26 26.0%	11 11.0%	70.3% 29.7%
2	11 11.0%	52 52.0%	82.5% 17.5%
	70.3% 29.7%	82.5% 17.5%	78.0% 22.0%

Fig 6.8. Confusion matrix for overall data

Table 6.1 Binary classification results for training, testing and validation data

Process	*TP*	*TN*	*FP*	*FN*	*Sensitivity*	*Specificity*	*Accuracy*
Training	*17*	*43*	*9*	*6*	*0.739*	*0.82*	*80%*
Testing	*5*	*4*	*1*	*0*	*1*	*0.80*	*90%*
Validation	*4*	*5*	*1*	*5*	*0.44*	*0.83*	*60%*
Overall	*26*	*11*	*52*	*11*	*0.70*	*0.17*	*78%*

A Receiving Operating Characteristic Graph (ROC) curve is the most commonly used way to visualize the performance of a binary classifier. The ROC curve for training data is shown in Fig.6.9. Fig 6.10 represents the ROC curve for test data. Fig.6.11 shows the ROC curve for validation data. The overall ROC curve is shown in Fig. 6.12.



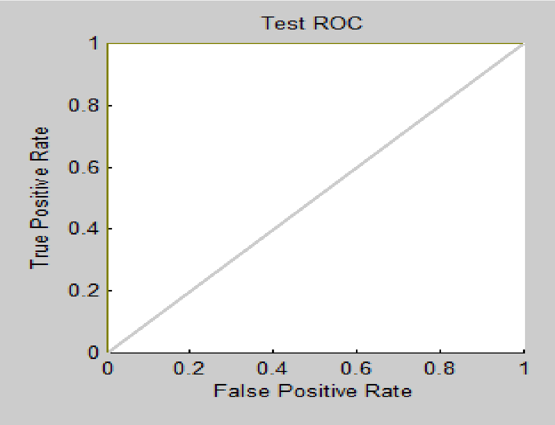
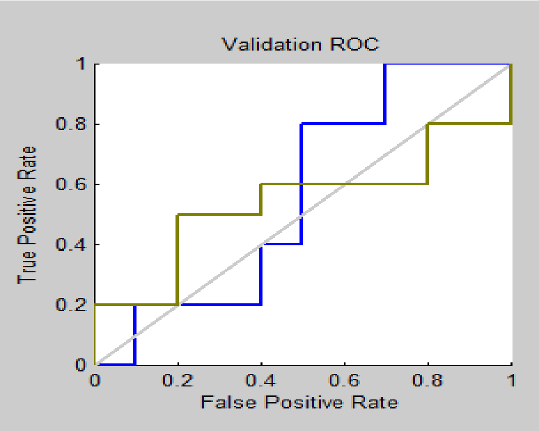


Fig 6.9.ROC curve for training data

Fig 6.10. ROC curve for Test data



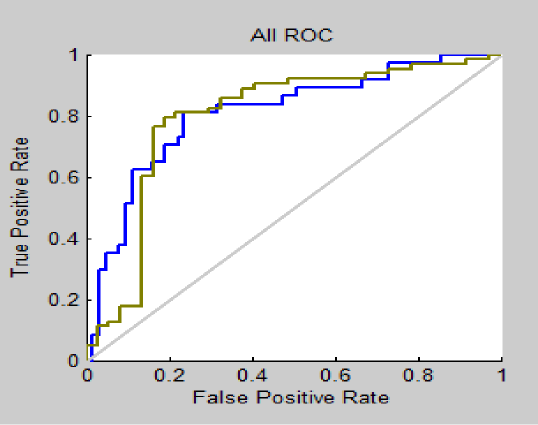


Fig.6.11. ROC curve for validation data

Fig.6.12. ROC curve for overall data.

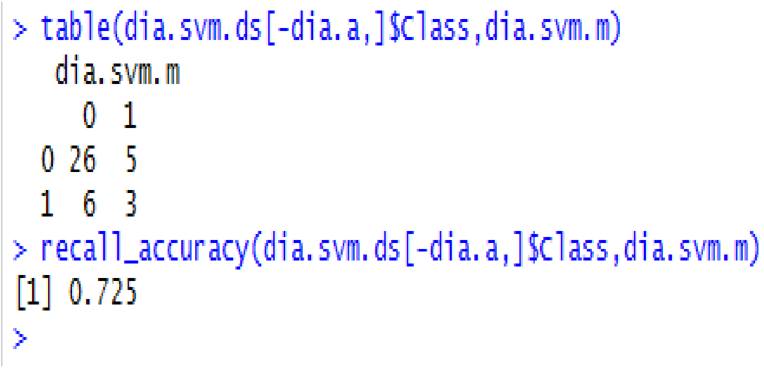
6.6 Results

Binary classification problem was designed to evaluate the performance of SVM and the accuracy of the test data was evaluated using R Studio.60% of the data was used for training and 40% of the data was used for testing. Radial Basis Function (RBF) kernel of SVM is used as classifier to train the model, as RBF kernel function is used for analyzing high dimensional data [69] . The RBF kernel function can be defined as,

$$K\ (x_i,\ x_j)=exp\ (-\gamma||x_i-x_j||^2, \quad \gamma>0 \qquad (6.6)$$

Where, γ is a kernel parameter and xi is a training vector. A larger value of γ will give a smoother decision surface and more regular decision boundary. The RBF with large γ will allow support vector to have a strong influence over large area. The value of γ is chosen as 0.5.

To evaluate the robustness of the SVM model a 10 fold cross validation was performed in the training data set to assess the quality of the model. The cost C- constant of the regularization was 4 and the value of ε (epsilon) is 0.1.epsilon is the insensitive-loss function. ε ,parameter is used for yielding better generalization performance[73] Tolerance of termination condition was taken as0.001,which is the default value. The cost parameter is used to avoid over fitting. C is a trade-off between the training error and the flatness of the solution. The larger the C, the less is the final error. But if the value of C is increased much, the generalization properties of the classifier may be lost. For RBF-SVM, two parameters need to be tuned: C and gamma. The method is to perform a good grid search. The goal is to identify good (C, γ),so that classifier can run accurately to predict the unknown data[74]. Tuning an object of class tune, includes the component best. Tune () which returns the best model detected by tune. The tune result returns the MSE. The process of choosing such parameters is called hyper parameter optimization, or model selection. Fig.3.13 shows the confusion matrix for the test dataset. The prediction accuracy of SVM was 72.5%. Fig.314. shows the plotting of test data of diabetes. The best performance was at MSE equal to 0.1888 and the best parameters were γ is 2 and cost is 8. The plot of grid search is shown in fig.3.15. On this graph we can see that the darker the region is, the better our model is (because the RMSE is closer to zero in darker regions).



```
> table(dia.svm.ds[-dia.a,]$Class,dia.svm.m)
   dia.svm.m
     0  1
  0 26  5
  1  6  3
> recall_accuracy(dia.svm.ds[-dia.a,]$Class,dia.svm.m)
[1] 0.725
>
```

Fig. 6.13. Confusion matrix of test data and prediction accuracy of SVM.

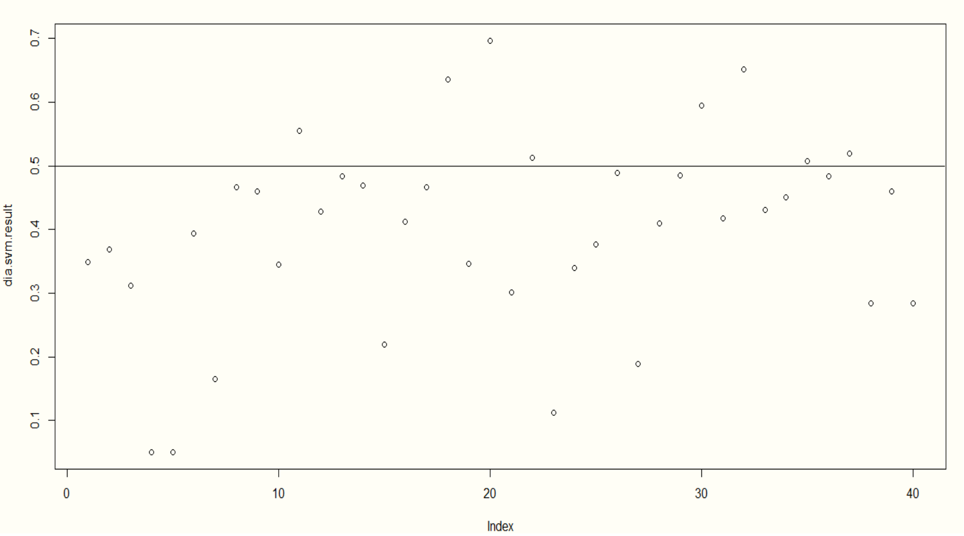


Fig.6.14. Plotting of test data.

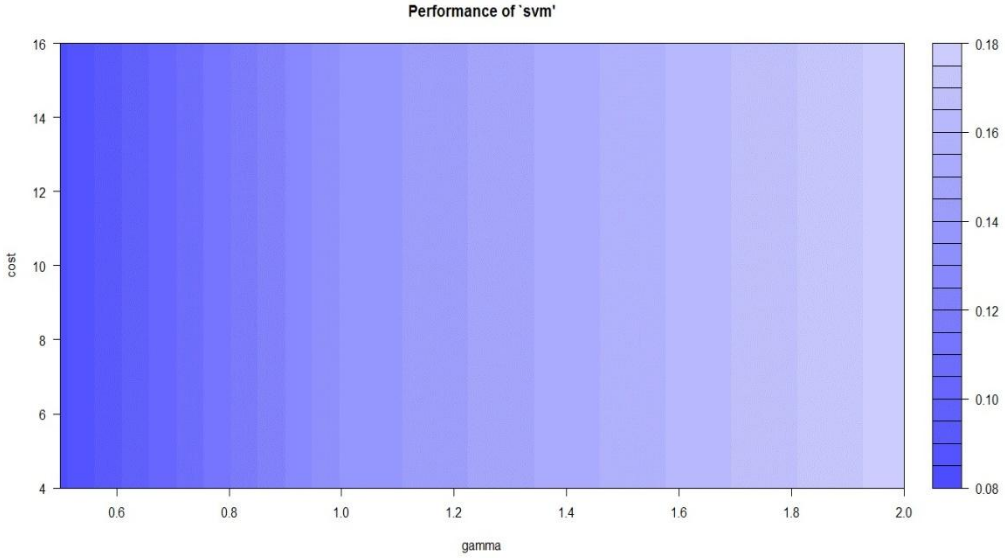


Fig.6.15. Grid search plot

The level of effectiveness of the classification model is calculated with the number of correct and incorrect classifications in each possible values of the variables being classified. From the results obtained from the confusion matrix, the following equations are used to measure the accuracy, sensitivity, specificity, misclassification rate, and prevalence.

$$Accuracy=TP+TN/TP+TN+FP+FN \qquad (6.7)$$

$$Sensitivity=TP/TP+FN \qquad (6.8)$$

$$Specificity=TN/TN+FP \qquad (6.9)$$

$$Misclassification\ rate=FP+FN/Total \qquad (6.10)$$

TP (True Positive): The number of examples correctly classified to that class.

TN (True Negative): The number of examples correctly rejected from that class.

FP (False Positive): The number of examples incorrectly rejected from that class.

FN (False Negative): The number of examples incorrectly classified to that class.

Table 6.2 Performance of SVM classifier

Test data	***TP***	***TN***	***FP***	***FN***	***Sensitivity***	***Specificity***	***Prevalence***	***AUC***	***Misclassification rate***	***Accuracy***
40	*3*	*26*	*5*	*6*	*0.333*	*0.838*	*0.225*	*0.686*	*0.275*	*72.5%*

ROC Curve is a commonly used graph that summarizes the performance of a classifier overall possible thresholds. It is generated by plotting the True Positive Rate (y-axis) against the False Positive Rate (x-axis) by varying the threshold for assigning observations to a given class. The ROC curve of SVM is shown in fig.3.16. The testing set accuracy of the SVM classifier is shown infig.6.17.

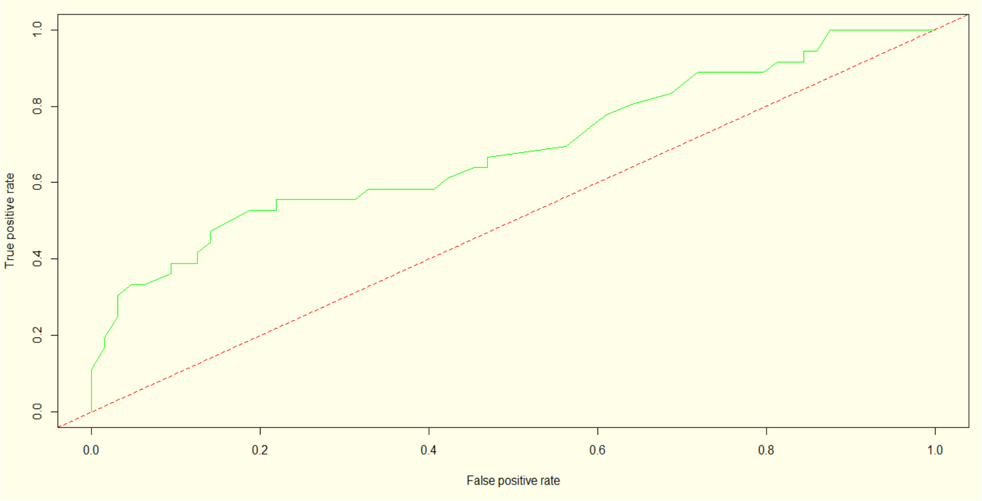


Fig. 7.16. ROC curve of SVM

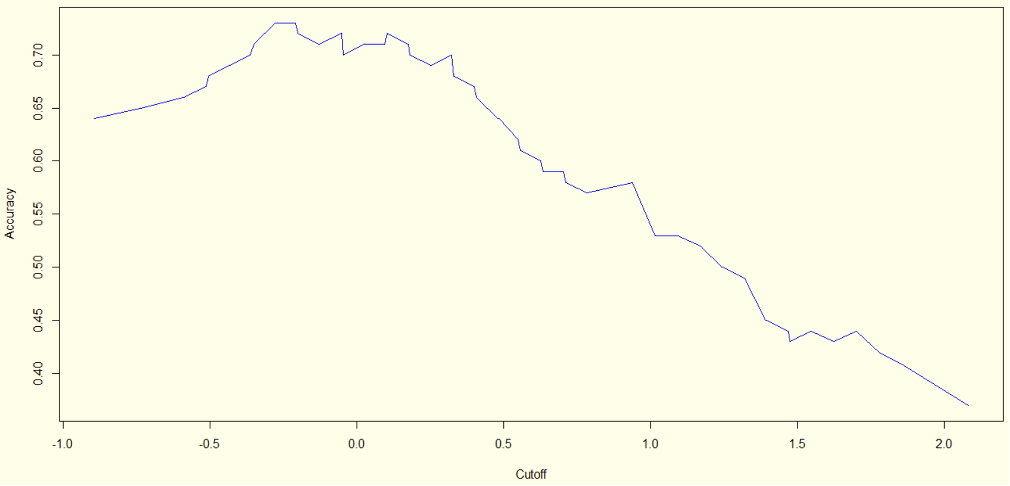


Fig. 7.17. Accuracy plot of SVM