Support Vector Machine Experiment Report

Manikanta Kalyan Gokavarapu
Department of Computer Science and Engineering
University at Buffalo, Buffalo, NY 14260
mgokavar@buffalo.edu

1 Support Vector Machine

The main objective of a support vector machine is to find a hyperplane in a N-dimensional space (Here N corresponds to number of features) that distinctly classifies the data points. Apart from a Hyperplane a support vector machine algorithm also creates two marginal lines with some distance (margin) between them to make sure that the data is easily linearly separable between the two classification points. These marginal lines are also hyperplanes that are parallel to the main hyperplane and one hyperplane passes through one of the nearest positive points and the other hyperplane or marginal line passes through one of the nearest negative points as shown in the figure [2.1].

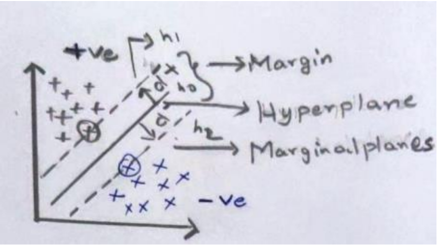


Figure [2.1]

The distance between these two parallel hyperplanes is known as margin. So, we can classify the data into a particular class by using these hyperplanes, like any data that is above the hyperplane ‘h1’ is classified into a positive data point and any point below the hyperplane ‘h2’ is classified as negative data point. The marginal distance between the hyperplanes is very important because it gives us a cushion to classify the data points more accurately. So, our main goal in a support vector machine is to select a best hyperplane that has maximum marginal distance. Maximizing the marginal distance provides some reinforcement so that future data points can be classified with more confidence. After finding the hyperplanes with maximum marginal distance we can get the support vectors. Support vectors are basically the points that are passing through the marginal hyper planes h1 and h2. These all SVM concepts can be applied to a linearly separable data but there is one more type of data known as Nonlinear Separable data. So, linearly separable data is the one in which we can easily separate the data using a straight line. But in cases of nonlinear separable it is not possible to divide the data using a straight line. So SVM makes use of a concept called SVM kernels. The main aim of these kernels is to convert low dimension data into high dimensional data so that we will be able to easily classify the data in higher dimensions using the hyperplanes.

Let’s see the basic math behind a linear SVM. So, suppose we have a main hyperplane ‘h_0’ with equation as ‘$w^Tx+b=0$’. From this hyperplane if we extend and go to the nearest data point of each class, we can find the negative and positive marginal planes. So, the equation of negative marginal plane ‘h_1’ is given as ‘$w^Tx+b=-1$’and positive marginal plane ‘h_2’ is given as ‘$w^Tx+b=+1$’ as shown in the figure [2.2].

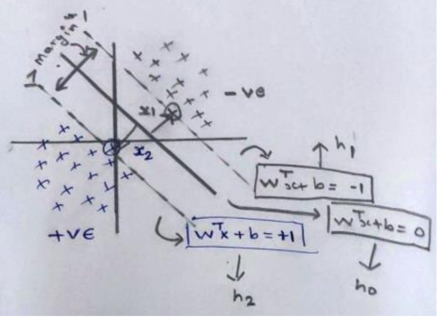


Figure [2.2]

In SVM as we need to consider maximum marginal distance hyperplane, we compute the distance between the points on the marginal hyperplanes h$_1$ and h$_2$ which is represented with the below equation.

$$\frac{w^T}{||w||}(x_2 - x_1) = \frac{2}{||w||}$$

Here 2/||w|| is known as "optimization function" and we need to maximize the value. So the whole optimization function is represented as below w.r.t hyper planes equations.

$$(w^*, b^*) \max \frac{2}{||w||} \quad st\ y_i \begin{cases} +1 & w^T x + b \geq 1 \\ -1 & w^T x + b \leq 1 \end{cases}$$

If We further compute the optimization formula for SVM is as below.

$$minimize_{w,b} \frac{1}{2}\ ||W||^2\ subject\ to\ y_i(w^T x_i + b) \geq 1\ where\ i = 1,2, \ldots\ldots, N$$

So, the final generalized formula of a linear SVM classifier would be as below by taking errors also into consideration.

$$(w^*, b^*) = \ \min \frac{||w||}{2} + C_i \sum_{i=1}^{N} Z_i$$

The C value in the above equation represents the number of errors a model can consider.

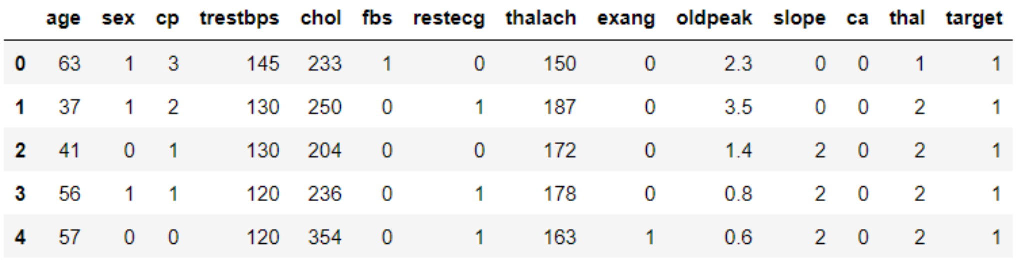
$\sum_{i=1}^{N} Z_i$ represents the summation (or) value of the error. The combined multiplication is called as regularization

2 Experiment

For the Experiment purposes I have taken a dataset which contain 14 attributes. The attributes data is as follows.

- Age - age in years
- Sex -1=male, 0=female
- Cp - chest pain type
- Trestbps - resting blood pressure (in mm Hg on admission to the hospital)
- Chol - serum cholesterol in mg/dl
- Fbs - (fasting blood sugar > 120 mg/dl) (1=true; 0=false)
- Restecg – resting electrocardiographic results
- Thalach – maximum heart rate achieved.
- Exang – exercise induced angina (1=Yes; 0=No)
- Oldpeak – ST depression induced by exercise
- Slope – the slope of the peak exercise ST segment
- Ca – number of major vessels (0-3) colored by fluoroscopy
- Thal – 3 = normal; 6= fixed defect; 7= reversable defect
- Target - have disease or not (1=yes, 0=no)

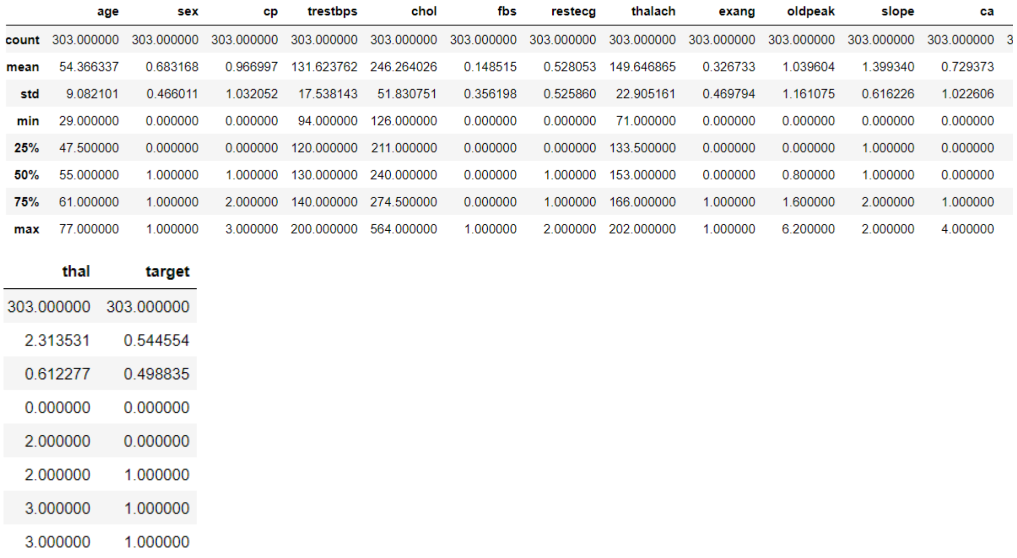
The first five rows of the dataset are given in the figure [2.3]



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Figure [2.3]

And the attribute information such as count, mean, std, min of the dataset is given in the figure [2.4].

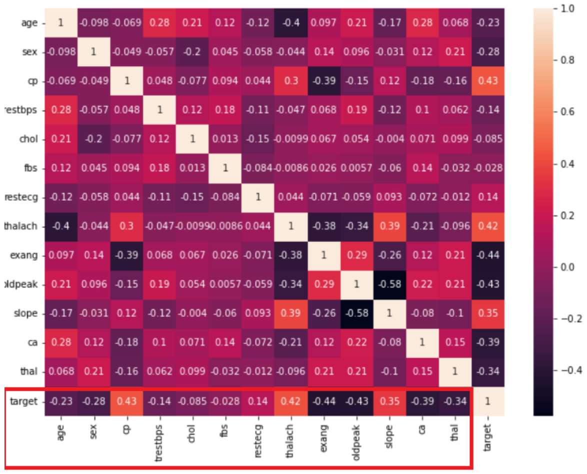


	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	3
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	

thal	target
303.000000	303.000000
2.313531	0.544554
0.612277	0.498835
0.000000	0.000000
2.000000	0.000000
2.000000	1.000000
3.000000	1.000000
3.000000	1.000000

Figure [2.4]

So, As I want to build a Support vector machine model to classify the people who will have heart disease or not (target) using some input features. First, I need to choose the input parameters or features to classify. Based on the correlation matrix output I have chosen all the features, as all the features are co-related with the target attribute to classify whether a person will have heart disease or not and difference between the correlated values is also minimal in the matrix. The correlation matrix output is given the below figure [2.5].



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
age	1	-0.098	-0.069	0.28	0.21	0.12	-0.12	-0.4	0.097	0.21	-0.17	0.28	0.068	-0.23
sex	-0.098	1	-0.049	-0.057	-0.2	0.045	-0.058	-0.044	0.14	0.096	-0.031	0.12	0.21	-0.28
cp	-0.069	-0.049	1	0.048	-0.077	0.094	0.044	0.3	-0.39	-0.15	0.12	-0.18	-0.16	0.43
restbps	0.28	-0.057	0.048	1	0.12	0.18	-0.11	-0.047	0.068	0.19	-0.12	0.1	0.062	-0.14
chol	0.21	-0.2	-0.077	0.12	1	0.013	-0.15	-0.0099	0.067	0.054	-0.004	0.071	0.099	-0.085
fbs	0.12	0.045	0.094	0.18	0.013	1	-0.084	-0.0086	0.026	0.0057	-0.06	0.14	-0.032	-0.028
restecg	-0.12	-0.058	0.044	-0.11	-0.15	-0.084	1	0.044	-0.071	-0.059	0.093	-0.072	-0.012	0.14
thalach	-0.4	-0.044	0.3	-0.047	-0.0099	-0.0086	0.044	1	-0.38	-0.34	0.39	-0.21	-0.096	0.42
exang	0.097	0.14	-0.39	0.068	0.067	0.026	-0.071	-0.38	1	0.29	-0.26	0.12	0.21	-0.44
ldpeak	0.21	0.096	-0.15	0.19	0.054	0.0057	-0.059	-0.34	0.29	1	-0.58	0.22	0.21	-0.43
slope	-0.17	-0.031	0.12	-0.12	-0.004	-0.06	0.093	0.39	-0.26	-0.58	1	-0.08	-0.1	0.35
ca	0.28	0.12	-0.18	0.1	0.071	0.14	-0.072	-0.21	0.12	0.22	-0.08	1	0.15	-0.39
thal	0.068	0.21	-0.16	0.062	0.099	-0.032	-0.012	-0.096	0.21	0.21	-0.1	0.15	1	-0.34
target	-0.23	-0.28	0.43	-0.14	-0.085	-0.028	0.14	0.42	-0.44	-0.43	0.35	-0.39	-0.34	1

Figure [2.5]

To build the Support vector machine model I have used a linear SVC kernel and to classify the data I have split the data into two parts in which 70% I used for train data and remaining 30% I used for test data. The test results are in below figure [2.6] plotted via confusion matrix and from the confusion matrix we can infer that 34 patients were predicted that they will not have heart disease, the prediction was correct (True-negative). 47 patients were predicted that they will have heart disease, the prediction was correct (True-positive). 6 patients were predicted that they will have heart disease, but the prediction was wrong (False-positive). 4 patients were predicted that they will not have heart disease, but the prediction was wrong (False-Negative). As False-Negative value is very less our model is performing pretty good better the back propagation neural network case that we performed earlier.

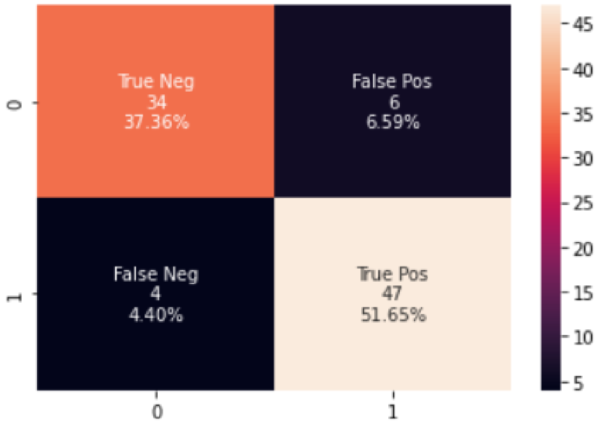
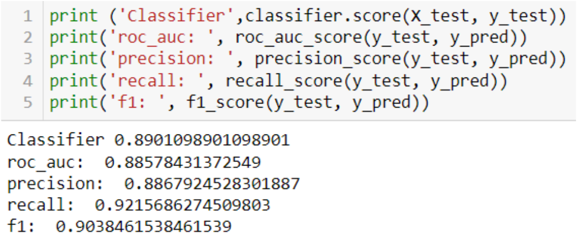


Figure [2.6]

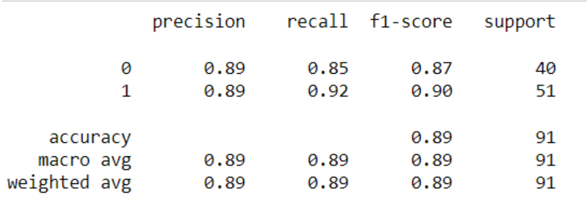
The overall accuracy or classifier score of the model is given as 89%. And remaining results of model like Roc_Auc score, precision score, recall score, F1 score are given in the below figure [2.7]. And the classification Report is given in the figure [2.8]



```
print ('Classifier',classifier.score(X_test, y_test))
print('roc_auc: ', roc_auc_score(y_test, y_pred))
print('precision: ', precision_score(y_test, y_pred))
print('recall: ', recall_score(y_test, y_pred))
print('f1: ', f1_score(y_test, y_pred))
```

```
Classifier 0.8901098901098901
roc_auc:  0.885784313725490
precision:  0.8867924528301887
recall:  0.9215686274509803
f1:  0.9038461538461539
```

Figure [2.7]



```
              precision    recall  f1-score   support

           0       0.89      0.85      0.87        40
           1       0.89      0.92      0.90        51

    accuracy                           0.89        91
   macro avg       0.89      0.89      0.89        91
weighted avg       0.89      0.89      0.89        91
```

Figure [2.8]

References

[1] Rohit Gandhi, Introduction to machine learning algorithms, 07/06/2018

[2] Charan kakararaparthi, Heart_Diseases, Prediction of Heart Disease by Effective Machine Learning techniques.

[3] Changyou Chen, Lecture Slides, Support Vector Machines, 04/10/2022.