An empirical study of Support Vector Machine and Naïve Bayes Classifier using Python and R

Dr. Gopal Pardesi[1], Nikita Pardeshi[2]

[1]Associate Professor, Department of IT, Thadomal Shahani Engineering College
[2]Final-Year B.E.-EXTC, Thadomal Shahani Engineering College

---***---

Abstract - Support Vector Machine classifier constructs a maximized hyper-plane that distinctly classifies the data points whereas, a Naïve Bayes classifier classifies data points using the Bayes' Theorem. This paper elaborately compares Support Vector Machine and Naïve Bayes classifiers using Python and R to classify tumors into malignant or benign. This paper compares accuracies of both the classifiers by means of a confusion matrix. The paper also suggests ways to improve accuracy of the models by normalizing training and testing datasets and gives a clear perspective for choosing among the two Machine Learning classifiers.

***Key Words*: Classifiers, Normalization, Bayes Theorem, Regression**

1.INTRODUCTION

Machine learning is broadly classified into two major types: Supervised and Unsupervised learning; Supervised learning being the more popular method. In Supervised Learning, machine is provided with labeled input data samples for training purposes and a predicted output is generated. Considering input variables 'X' and an output variable 'Y', the goal of Supervised Learning is to map input data with the output data [1]. Any particular dataset is split into training and testing datasets using in-built functions or packages. Once training and processing of the model is done, it is tested using a sample data to check whether it is predicting the correct, exact output or not. Various accuracy metrics such as F-loss, Jaccard index, Log Loss can be used to calculate accuracy of the model created.

Supervised learning algorithms such as Support Vector Machines, Naïve Bayes, Decision Trees, Random Forest etc. have been proved to provide efficient models with high accuracies. [8]

Supervised Learning is further categorized into Classification and Regression. Classification, as the name suggests, is used to segregate data into mutually exclusive labels or classes based on some parameters present in the input data. For example, based on previously recorded results and other relevant parameters, we want to predict whether Team A will win a match or not, there will exist two labels: Yes or No. Classification may be binary or multi-class. SVM and Naïve Bayes algorithms can be used for both, binary and multi-class classification.

Regression analysis refers to studying the relationship between independent variable (X) and dependent variable (Y) [3]. The output variable is a real or continuous value. The simplest model is Linear Regression which tries to fit the data with the best hyper-plane, also called as the best fit line, which goes through a set of points. The model predicts a dependent variable value, such as salary of a person, based on a given independent variable, such as work experience [2]. Hypothesis function for linear regression:

$\hat{Y}$= slope* X + intercept

where $\hat{Y}$= predicted output variable,

X= input variable

A best fit line must be obtained such that the error difference between the predicted and true variable must be minimum. This error difference is also called Mean Squared Error (MSE) [2].

2. SUPPORT VECTOR MACHINE

The basic idea behind SVM is to maximize the classification boundaries that is, to maximize the hyperplane [4]. Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane [5]. Since in the SVM training set, only support vector, a subset of training points, is valid sample, so before training the classifier, extracting the support vector can effectively improve the time and space efficiency [4]. Support vector machines are effective in high dimensional spaces and also in cases where the number of dimensions is greater than the number of samples [6]. In cases where the number of features is greater than the number of samples, over-fitting may occur [6].

Python has an inbuilt library- scikit-learn containing the SVC function whereas, R has the package e1071 [4] offering quick and easy implementation of both SVM and Naïve Bayes. The above-mentioned packages and libraries are discussed later in the paper.

SVM classifiers are most effective when there is a clear boundary of separation between classes. Another scenario where SVM works effectively is when the number of positive samples or records are about the same number as that of negative samples in the dataset. It is a cost-effective, memory efficient and simple classification method in machine learning. [10]

3. NAÏVE BAYES CLASSIFIER

Naïve Bayes' Classifier technique is based on the Bayesian theorem wherein likelihood or probability of occurrence of any event is calculated [7]. Any Naïve Bayes' classifier assumes strong independence between features.

$$P(A|B) = \frac{P(A)*P(B|A)}{P(B)}$$

Using Bayes Theorem, we can find probability of A happening, given that B has occurred. Here, B is the evidence and A is the hypothesis.

There exist three types of Naïve Bayes Classifiers:

a) Multinomial Naïve Bayes- used mostly for document classification problem to check the category which a document belongs to.
b) Bernoulli Naïve Bayes- Here, predictors are Boolean variables i.e. parameters we use to predict the class variable take up only two values, for example, zero or one, yes or no etc.
c) Gaussian Naïve Bayes- Predictors take up a continuous value i.e. they are not discrete in nature. Hence, we assume that these values are sampled from a Gaussian or Normal Distribution. [7]

4. DATA SET

The Wisconsin Breast Cancer data set comprises records of 699 patients with a total of 11 different variables. Depending upon values of these variables, we can predict if a tumor is benign (B) or malignant (M).

Classes: benign, malignant

Class distribution: Benign- 458(65.5%)

Malignant- 241 (34.5%)

5. EVALUATION IN PYTHON

5.1 CREATING SVM AND NAÏVE BAYES MODEL

After loading required libraries and data set in Python, we create a 'target' variable, as shown in Figure 2, which contains Boolean values: 0 for Malignant (M) and 1 for Benign (B) samples.

After dividing the data set into training and testing data, we fit the training data set into respective models: SVM and Naïve Bayes.

Prediction is made using the predict() function and a confusion matrix is plotted. The confusion matrix is a powerful visualization tool to visualize performance of an algorithm. It has 4 parameters:

i. TP- Observation and prediction are both positive
ii. FN- Observation is positive but the predicted value is negative
iii. FP- Observation is negative, but predicted value is positive
iv. TN- Observation and prediction are both negative

5.2 RESULTS

The following results are obtained:

Table 1:- Confusion Matrix

	Predicted Cancer	Predicted Healthy
Is cancer	290	0
Is healthy	165	0

This model has an accuracy of 64% only. In order to obtain a better model, we normalize the training and testing data set using the formula:

$$X(normalized) = \frac{X - Xmin}{Xmax - Xmin}$$

After normalization, we obtain the following accuracies:

Table 2:- Accuracy measures

Model	Accuracy	Specificity
SVM	95.1%	0.93
Naïve Bayes	95.54%	1

Specificity answers the following question: Of all the people who are healthy, how many of those did we correctly predict? *Specificity = TN/(TN+FP).*

6. EVALUATION IN R

The purpose of including R language in this paper is because R is easier, effective and includes a rich variety of 15,335 packages available in open-source repository [4].

For this specific project, using R is extremely useful since:

i. Accuracy score is higher for the models as compared to those obtained in Python
ii. Readily available libraries for each task

6.1 CREATING SVM AND NAÏVE BAYES MODEL

Different packages utilized are:

- caTools- required for splitting the data set into training and testing data
- e1071- provides training function for SVM and Naïve Bayes
- caret- provides a number of methods to estimate the accuracy of a machine learning algorithm [4].

After loading the data set, we split the data set into training and testing data and fit this into the SVM model and Naïve Bayes model using svm() and naiveBayes() functions, respectively.

6.2 RESULTS

Next, confusionMatrix() function is used and following results are obtained in R:

- Accuracy and Specificity for SVM model in R

Accuracy: 0.9789

95% CI: (0.9395, 0.9956)

No Information Rate: 0.6268

P-Value [Acc > NIR]: <2e-16

Kappa: 0.9543

Sensitivity: 1.0000

Specificity: 0.9434

- Accuracy and Specificity for Naïve Bayes model in R

Accuracy: 0.9366

95% CI: (0.8831, 0.9706)

No Information Rate: 0.6268

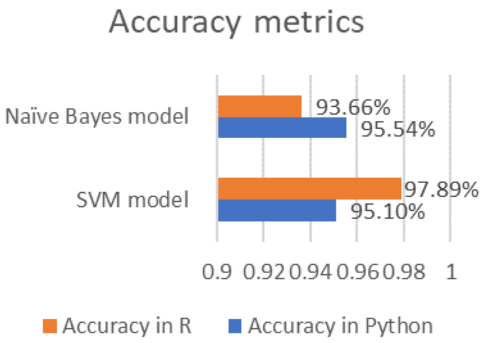
P-Value [Acc > NIR]: <2e-16

Kappa: 0.8619

Sensitivity: 0.9775

Specificity: 0.8679

7. CONCLUSION



REFERENCES

[1] Sheena Angra, Sachin Ahuja, 'Machine Learning and its Applications: A review', International Conference on Big Data Analytics and Computational Intelligence, October 2017

[2] Shen Rong, Zhang Bao-wen, 'The research of regression model in machine learning field', MATEC Web of Conferences 176, 01033 (2018)

[3] Zuyu Yin, Jianxing Liu, Minjia Krueger, Huijun Gao, 'Introduction of SVM algorithms and recent applications about fault diagnosis and other aspects', 2015 IEEE 13th International Conference on Industrial Informatics (INDIN)

[4] Packages in R: https://cran.r-project.org/web/packages/e1071/e1071.pdf

[5] Megha Rathi, Arun Kumar Singh, 'Breast Cancer Prediction using Naïve Bayes Classifier', International Journal of Information Technology & Systems, Vol. 1; No. 2: ISSN: 2277-9825 (July-Dec. 2012)

[6] Sivapriya J, Aravind Kumar V, Siddarth Sai S, Sriram S, 'Breast Cancer Prediction using Machine Learning', International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-4, November 2019

[7] Feng-Jen Yang, 'An Implementation of Naïve Bayes Classifier', 2018 International Conference on Computational Science and Computational Intelligence (CSCI)

[8] Himani Bhavsar, Mahesh H. Panchal, 'A Review on Support Vector Machine for Data Classification', International Journal of Advanced Research in Computer Engineering and Technology (IJARCET), Volume 1, Issue 10, December 12

[9] Guosheng Wang, 'A Survey on Training Algorithms for Support Vector Machine Classifiers', 2008 Fourth International Conference on Netweoked Computing and Advanced Information Management, September 12

[10] Shubham Sharma, Archit Aggarwal, Tanupriya Choudhary, 'Breast Cancer Detection using Machine Learning Algorithms', 2018 Inernational Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS), July 25, 2019