Training Machine to Classify Severity of Diseases

Kirtika Yadav#, Reema Thareja
Department of Computer Science, University of Paderborn, Paderborn, Germany,
Department of Computer Science, SPM College, University of Delhi, India
#Corresponding Author, Email: kirtika.yadav9@gmail.com

***Abstract*—With the invention and usage of new technologies, the need for storing data of users has increased manifolds. Nowadays, the amount of data is increasing, so it is time consuming to look at every feature of the data set to discriminate different objects based on some features that are necessary to classify them. It becomes so important for companies to analyze their data and using classification that can give an accurate result of data that support in business decisions which in result help companies to lead in the cooperation world. Thus, Naive Bayes is one of the most popular probabilistic classification algorithms that is adopted by top companies for the classification task. In this paper, our focus is to use Naive Bayes algorithm to classify severity of cancer in patients based on their symptoms.**

***Keywords*—Bayes theorem, Classification, Data mining, Machine learning, Naive Bayes, Naive rule.**

1. INTRODUCTION

In mathematical theory, we calculate the probability that an event will occur. We can even calculate occurrence of group events that occur together. There is another concept of probability in which we can calculate the probability of occurrence of another event after some event already happened. Data science uses this concept of probability by utilizing the probability that an event has already occurred and then calculates the proportion of where the second event will occur. That is why the Naive Bayes classification is introduced which can predict about the outcome by using the training dataset. But before understanding the concept of Naive Bayes it is required to first understand the concept of Bayes algorithm as Naive Bayes is based on the formula of Bayes theorem.

The organization of this paper is like: section 1 deals with the introductory part of the probability concept that is used in Naive Bayes. The related work is discussed in section 2. In section 3, we have given brief description about Bayes Theorem and how it is used is Naive Bayes Classifier. Section 4 discusses the implementation of the classification techniques along with the results. Section 5 discusses about the advantages and disadvantages of the Naive Bayes. In section 6 the final the result is showed and in last the paper is finally concluded.

2. RELATED WORKS

A study carried out by T. John Peter and K. Somasundaram [19] that used different methods to improve the classification methods by checking the significance of feature selection algorithms. They implemented a feature selection algorithm by combining CFS and Bayes theorem concepts and after performing this algorithm by using the health care domain dataset, they got 85.5% accuracy and also found that this method removes more irrelevant attributes and also improves the classifiers performance. This feature selection method is used with four algorithms (Naive Bayes, J48, KNN algorithm and Multilayer Perception) for better accuracy result. This method gave good accuracy for Naïve Bayes and KNN classifiers.

Many researchers have been using different classification techniques that can be useful for the diagnosis of heart disease. One of the best techniques is Naive Bayes that is considered more useful in diagnosis of heart diseases. Mai Shouman used the dataset with 297 rows for checking whether the integration of K-means with Naive Bayes is successful in the diagnosis of the heart disease [20]. This paper used different initial centroid selection issue that affects the K means clustering. As a result, it was shown that this integration technique improved the Naive Bayes accuracy which was 84.5% for the heart disease patient's diagnosis.

In a study, researcher Ayman M Mansour performed the texture classification by using Naive Bayes for classification and Independent Component Analysis (ICA) for extracting texture features as it was able to correctly capture the detailed information of textured images and also label them correctly [21]. Use of Naive Bayes also improved the classification error rate which made the Naïve Bayes Classifier much more useful in texture classification. The experiment was performed by using 5640 texture from the dataset, and Naive Bayes outperformed other classifiers (ICA, SVM, Wavelet, and Gaber Filter) with 99.4% accuracy.

Another study that is performed by S Vijayarani and S Deepa who also used Naive Bayes classifier for prediction of disease using the protein sequences which were obtained from the patients [22]. Naive Bayes is used to improve the accuracy of classification between normal and diseased protein sequence. Accuracy rate obtained with 20 and 50 instances of dataset were 84% and 85%. It was shown in the paper that Naive Bayes proved to be proficient for detecting disease sequence with good accuracy.

3. Bayes Theorem

In machine learning, Bayes theorem provides a way to calculate the probability of a prior event (meaning an event is occurring), given that another subsequent event has occurred. Bayes theorem is a method that is used for predicting values, that is, it gives us a formula for computing the probability that a certain record belongs to a given class, given with record's attribute [1]. It only makes prediction on the basis of prior knowledge (meaning, probable guess on an outcome without any information about its attribute) [2]. But with the addition of more evidence, the prediction made by the theorem also changes which is important for the theorem for classification [3]. Now, suppose we have m classes C1, C2………Cm along with the predictors value which are given as X1, X2…Xp and the probability of the given classes are P(C1), P(C2)……P(Cm). Now, the main task is to classify a record by using the predictors. If we know the probability of occurrence of X1, X2…….Xp within each class which is represented as P(X1),P(X2)……P(Xp),then the Bayes theorem formula for the calculation of the probability of class Ci given Xp is given as [1]:

$$P\ (Ci|X1\ldots\ldots,Xp) = \frac{P(X1,\ldots\ldots Xp|Ci)P(Ci)}{P(X1\ldots\ldots.Xp|C1)P(C1)+\cdots\ldots.P(X1\ldots\ldots.Xp)|Cm)P(Cm)} \quad (1)$$

We can write the bayes formula as follows [4]:

$$P\ (A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (2)$$

Here, A and B are events where B is an evidence and according to this equation, we are calculating the probability of the event A given that B is true. By the word evidence we mean an attribute value of an unknown instance [4].

P (A) is the prior probability (probability of an event without any evidence is seen) and P (A|B) is the posterior probability (probability when evidence is seen) and P (B|A) is said as conditional probability of B given A [4]. Our main concern is to calculate the posterior probability of P (A|B). Now, it is required to classify the data to the class that holds the highest probability value. This is also referred as maximum posteriori which can be written as

$$MAP\ (A) = \max\ (P\ (A|B))\ \text{OR}\ MAP\ (A) = \max\ (P\ (B|A) * P\ (A)) \quad (3)$$

3.1 Naive Bayes Classifier

Before moving to the concept of the Naive Bayes we first need to know what is the naive rule used in classification.

Naive Rule**:** For the classification of data in one of m classes, by ignoring the predictor information the naive rule is used to classify the data as a member of the majority class. Naive rule is basically used for the calculation of the working of complicated classifier.

Naive Bayes: The Naive Bayes classifier is one of the simplest still more sophisticated techniques of classification which combine the set of given predictors information into the naive rule in order to get accurate result of classification [1]**.** That is, the probability of a variable belonging to a class can be obtained not only by using the prevalence of that class but also on the basis of additional knowledge on that data. This method has a set of supervised learning algorithms that is based on Bayes' Theorem with an assumption of independence among predictor [5]. It means that the features in a class are independent of any other features. That is why this classifier is called Naive Bayes classifier. Naive Bayes classifier can be used for both continuous and categorical variables. If numerical value is given, it must be put aside and must convert it into categorical value before Naive Bayes algorithm works on it. If a large dataset is given, then it is recommended to use Naive Bayes classifier. Google which is the top most web searching company also use Naive Bayes classifier in spelling correction when a user type in the search bar. Whenever a user type wrong spelling of a word then the Google indicate the correct spelling for that word.

Naive Assumption: The assumption made in the Naive Bayes is that the pair of features is independent of each other. So, we divide the evidence into different independent attributes. This independent feature assumption is not always correct but sometimes works fine in practice [6].

It is based on the Bayes formula that is probability of event A given B evidence which is given above also as [7]:

$$P(A,B)=P(A)\ P(B) \qquad (4)$$

Through this formula and using the concept of bayes theorem (from equation (2)) the final formula is as follows:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \qquad \text{(from (2))}$$

Here, A is a class and B is an instance. (A) represents the dependent events which means a predicted variable and (B) represents the prior event which means a predictor attribute. This Bayes formula is already explained above in equation 1.

Let's take an example for better understanding of the formula of Naive Bayes classifier. I used the example of fruit prediction. The size of the training dataset is 1200 fruits. The attribute set of the dataset are as follows: Fruit {Yellow, Sweet, long}, we have a fruit which is red, sweet and round and we have to classify the class to which the fruit belong to by using the Naive Bayes algorithm which is based on probability concept of Bayes theorem. Also, the data set has three different classes which are: apple, banana, and others.

Frequency table of the dataset is given in table 1 as:

Table 1. Frequency table of three different classes: APPLE, BANANA, and OTHERS.

FRUIT	RED	SWEET	ROUND	TOTAL
Apple	400	300	350	400
Banana	350	450	0	650
others	50	100	50	150
total	800	850	400	1200

Now, the formula for the calculation of probability using Naive Bayes which is based on Bayes theorem is given in equation (2) which is written as

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

Here, P (A) = Class prior probability

P (B|A) = Likelihood prior probability

P (B) =Predictor prior probability

This above formula can be visualized as

Posterior probability = (prior probability* likelihood)/P(evidence), that is probability of predictor prior

So, first we will calculate the class prior probability for red, sweet and round fruit.

P (red) = 800/1200 =0.66

P (sweet) =850/1200=0.70

P (round) =400/1200=0.33

Now we will calculate the likelihood prior probabilities which are as follows in table 2:

Table 2. Calculation of likelihood prior probabilities

For Apple	For Banana	For Other
P(apple|red)=400/800=0.5	P(banana|red)=350/800=0.43	P(others|red)=50/800=0.062
P(apple|sweet)=300/850=0.35	P(banana|sweet)=450/850=0.52	P(others|sweet)=100/850=0.11
P(apple|round)=350/400=0.85	P(banana|round)=0/400=0	P(others|round)=50/400=0.125

Now the likelihood table for the given features and the classes is shown in table 3:

Table 3. The likelihood result of the three classes

FRUIT	RED	SWEET	ROUND	TOTAL
Apple	P(apple|red)= 0.5	P(apple|sweet)= 0.35	P(apple|round)= 0.85	P(apple)=400/1200=0.33
Banana	P(banana|red) =0.43	P(banana|sweet) =0.52	P(banana|round)= 0	P(banana)=650/1200 =0.54
Others	P(others|red)= 0.062	P(others|sweet)= 0.11	P(others|round)= 0.125	P(others)=150/1200=0.125
Total	800	850	400	1200

Now the next step is to calculate the conditional probability by using equation (2) and equation (4) of different classes to classify which fruit it is which is red, sweet, round [8][9].

$$P(apple|red,sweet,round) = \frac{P(apple|red)* P(apple|sweet)* P(apple|round)*P(apple)}{P(red,sweet,round)}$$

$$= \frac{0.5*0.35*0.85*0.33}{P} = 0.21$$

$$P(banana|red,sweet,round)= \frac{P(banana|red)* P(banana|sweet)* P(banana|round)*P(banana)}{P(red,sweet,round)}$$

$$=\frac{0.43*0.52*0*0.54}{P}=0$$

P(others|red,sweet,round)=

$$\frac{P(\text{others}|\text{red}) * P(\text{others}|\text{sweet}) * P(\text{others}|\text{round}) * P(\text{others})}{P(red,sweet,round)}$$

$$=\frac{0.062*0.11*0.125*0.125}{P} = 0.09$$

Now the final step of naive bayes algorithm is to find the maximum probability [6]

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y) \qquad (5)$$

In the above example the apple class has the highest probability i.e., (0.21> 0.9). This shows that the fruit which is red, sweet, and round belongs to apple fruit class. But one problem is that what if the probability of a feature is zero.

Laplace Smoothing: In the above example, the probability for P (banana| red, sweet, round) was zero

$$= \frac{P(\text{banana}|\text{red}) * P(\text{banana}|\text{sweet}) * P(\text{banana}|\text{round}) * P(\text{banana})}{P(red,sweet,round)}$$

$$=\frac{0.43*0.52*0*0.54}{P}=0$$

As naive classifier uses the multiplication of the conditional probabilities of feature on each class. If one of the terms is zero then multiplication with 0 gives the result zero. This means that we don’t get any information at all by doing this. Thus, a solution for such kind of problem introduces the concept of Laplace smoothing. The Laplace Smoother simply adds small counts in the frequencies of each feature to make sure that the features have non-zero probability for each class [10]. By adding 1 value will be sufficient to get non-zero probability.

The formulas that are mentioned above work perfectly on the discrete or categorical values. When dataset contain continuous data; some assumptions are needed for the classification of the features. There are some Naive Bayes classifiers that are useful for handling the continuous features depending on the distribution of P (Xj|Ci) [11].

3.2 Types of Naive Bayes Classifier

The Naive Bayes classifier is mainly used to predict the class that has categorical values only. But it is kind of difficult for the theorem to work perfectly on discrete values. Thus, there are variations in the Naive Bayes classifier that can handle different discrete values for the prediction which are as follows:

a. Multinomial Naive Bayes Classifier
b. Bernoulli Naive Bayes Classifier
c. Gaussian Naive Bayes Classifier

a. Multinomial Naive Bayes: The Multinomial Naive Bayes algorithm is used for document-based classification only like documents based on technology, sports etc. That is, this algorithm is used when features are multinomial distributed. The features used in this algorithm include the frequency of words that are present in the given document [9]. This algorithm works well when data can be easily counted like count of words in the text.

The formula for the likelihood calculation in multinomial naive bayes is given as [12]:

$$\text{P(X|Ck)}= \frac{(\sum_i Xi)!}{\prod_i Xi!} \prod Pki^{Xi} \qquad (6)$$

Here, Xi is a feature vector that counts the number of times i event occur in an instance.

*b. Bernoulli Naive Bayes Classifier***:** The Bernoulli Naive Bayes algorithm is used for the Boolean variables-based classification instead of the calculation of frequency of words. This algorithm used to take values yes or no for the prediction of the class only. This model is useful in those document classifications where binary features are used (that is, whether sports occur in a document or not). Naive Bayes classifier is used in some technologies like in spam filtration, adult content detection and so on [13]. The Bernoulli Naive Bayes classifier decision rule is based on a formula which is given as [12]:

$$\text{P (Xi|y) = P(i|y) Xi + (1-P(i|y)) (1-Xi)} \qquad (7)$$

This formula is different from the Multinomial Naive Bayes classifier rule as it clearly and completely expresses the non-occurrence of a feature i which act as an indicator for class y [14].

c. Gaussian Naive Bayes Classifier: If a dataset contains continuous values, not discrete values than Gaussian Naive Bayes classifier is used. The likelihood is taken as a Gaussian [12] [14]. Thus, the formula for the conditional probability using Gaussian distribution is given as [3]:

$$P\ (Xi|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(Xi-\mu_y)^2}{2\sigma_y^2}\right) \qquad (8)$$

Here, $\mu_y =$ is the mean of values in y related with a class Xi
σ_y^2 = variance of values in y related with a class Xi
Y= continuous attribute

4. Naive bayes classification for predicting cancer

As discussed theoretically above how the classification is performed by using Naive Bayes classifier. In this section, Naive Bayes classifier is illustrated using R language. For the implementation part, we have used Rstudio because it is easy to understand how to code in R and makes the task much more flexible than other tools.

For analysis, we have used the breast cancer dataset. This dataset is used to predict the stage of cancer of a cancer patient by understanding the features of the dataset. There are various packages that we can use for the implementation of Naive Bayes. Here, we used e1071 and caret. The e1071 package is used to provide functions of probability, for class analysis, naivebayes () function that makes the calculation for Naive Bayes classifier easy [15]. This package made the task for classification using Naive Bayes much simpler.

The dataset used for the classification consist of class diagnosis that has two levels “B” (benign) and “M” (malignant) which is used to predict about the class of diagnosis. Now we will begin by including the data set breast-cancer.csv and by using str() to view details about the dataset which is shown as below:

```
>str(breast-cancer)
Classes ‘spec_tbl_df’, ‘tbl_df’, ‘tbl’ and 'data.frame':569 obs. of 32 variables:
```

The dataset has 569 observations on 32 variables. In this the class variable is diagnosis that has two levels only “B” and “M”. The task of the Naive Bayes classifier is to calculate the posterior probability which is obtained from Bayes theorem and accordingly return the value of class which can be used for prediction for unlabeled features. Naive Bayes algorithm can work well with the large datasets. This is one of the techniques that can give the highest accuracy rate.Now it is required to first divide the dataset into training and test dataset which are required for the classification.

```
bc=read.csv(file.choose(),header=T)

train=bc[1:450,]

test=bc[451:569,]


bc%>%
ggplot(aes(x=compactness_mean,fill=diagnosis))+

geom_density(alpha=0.8,color='black')+
ggtitle("density plot")

test%>%
ggplot(aes(x=compactness_mean,fill=diagnosis))+

geom_density(alpha=0.8,
color='black')+ggtitle("density plot")

train%>%
ggplot(aes(x=compactness_mean,fill=diagnosis))+
geom_density(alpha=0.8,
color='black')+ggtitle("density plot")
```

After applying density plot function for original dataset, train dataset and test dataset the following graphs are represented as:

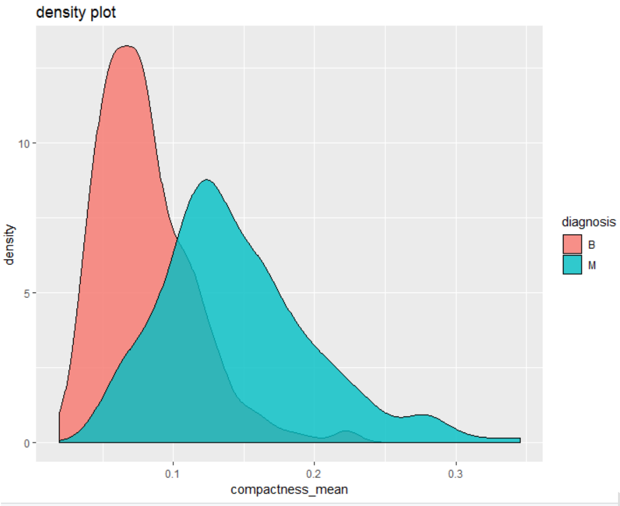

Fig. 1 Density plot for original dataset of breast cancer

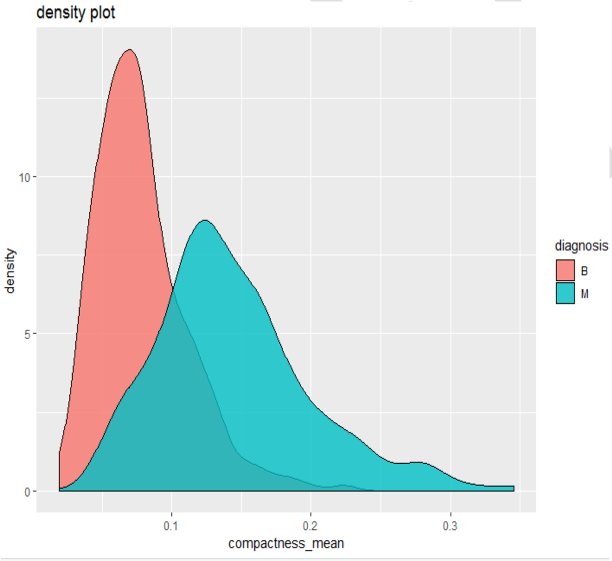

Fig. 2 Density plot for training dataset obtained from the original data set**.**

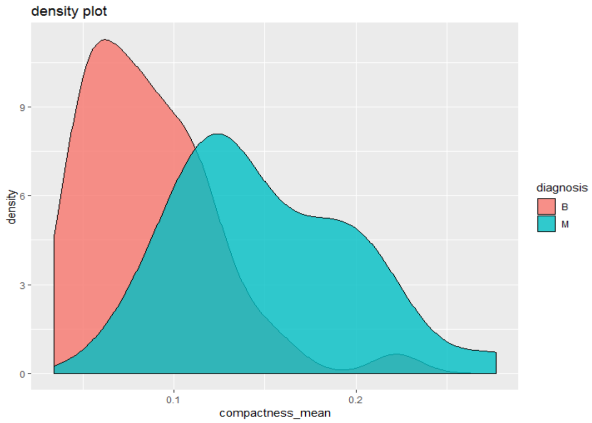

Fig. 3 Density plot for training dataset obtained from the original data set

We imported the dataset breast cancer using read.csv in bc variable. The nrow (bc) shows the number of rows in the dataset taken for classification. Here, crude splitting is performed on the dataset to create training dataset and test dataset. In this example, 450 elements are there in the training dataset and 119 elements in the test dataset. Now we build our model using training data file and test our model using test dataset. After that we need to check how many levels are present in the dataset. So, we used levels () function on our training dataset which showed two levels “B” and “M”. We used density plot to show which level of diagnosis has high compactness mean value and alpha=0.8 to show how transparent the graph is. By looking at all the three graphs we can see that “B” level has high compactness than the “M” level. There is a significant amount of overlap between the two levels. Next step is to apply the naïve bayes function which can be done as:

```
model=
naiveBayes(diagnosis~.,data=train,laplace
=1)

>model

Naive Bayes Classifier for Discrete Predic
tors

A-priori probabilities:
Y
     B       M
```

```
0.5888889 0.4111111
Conditional probabilities:
id
Y        [,1]       [,2]
  B 25418892 110139304
  M 30170444 115071376
radius_mean
Y        [,1]     [,2]
  B 12.11126 1.721049
  M 17.28103 3.153204
:
fractal_dimension_worst
Y          [,1]        [,2]
  B 0.07865143 0.01335498
  M 0.09184043 0.02176034
>class (model)
[1] "naiveBayes"

>model$apriori
Y
  B   M
265 185
```

In the above part as we used the e1071 library that holds the Naive Bayes classifier function which is naivebayes() that helps in performing Naive Bayes classification. It takes categorical value and table as input and returns an object which is further sent to predict function to predict the outcome of unlabeled feature. To create Naive Bayes model, we applied the Naive Bayes function along with the Laplace parameter. The Laplace is initialized with 1 as Naive Bayes is based on the calculation of conditional probabilities of each feature on each class. If any feature has zero probability of occurrence for a class, then it can lead to posterior probability to be zero for that class. To avoid this, we have included Laplace=1 in the above code. The naiveBayes() function also generated the conditional probability which is basically a likelihood calculation for each attribute in the dataset.

With the use of model$apriori we have represented the class distribution in the data set, that is, the prior distribution of the classes is represented here. As we can see there are 265 belongs to "B" class and 185 belongs to "M" class of the training dataset. And the a-prior probabilities that is obtained from naiveBayes() function are the prior probability. If any class is rare in the prior probability than that level will hardly occur in the test dataset [13]. Now it is required to perform the prediction function for the test dataset.

Predicting Values**:** In the above part the naivebayes () function is used which return the object of class which is displayed by class (model) and get "naiveBayes". Such object is used for the prediction function which can predict the outcome for unlabeled variables. It returns an object of class "naiveBayes". This object is used by predict () function for prediction of outcomes of unlabeled features. Now we can perform prediction using the predict function that uses the model for classification of observations on the basis of conditional probability that is obtained in above code. The predict function can be implemented as follows:

```
pred<-predict (model,test,type="class")
table (pred,test$diagnosis,dnn=c("Prediction","Actual"))
Actual
Prediction   B    M
B 85   2
M 725
```

In this part, the predict function () by default returns the class on the basis of the highest conditional probability for the prediction. This function helps us to specify whether we want the class with highest probability of occurrence or need to find probability for each class. If we want to check the conditional probability of each class, we can use type=" raw" instead of type="class". Here, predict function is used to predict the cancer stage for the test dataset. To display the confusion matrix (used for examining the accuracy of the model), we have used the table method. We could have also used the confusionmatrix() function by using caret library as shown below.

```
>install. packages ("caret")
```

```
>library (caret)
```

Now for the confusion matrix that returns the complete output along with the accuracy is shown below:

```
>cmatrix<-table(test$diagnosis,pred)

>plot (cmatrix)

>cfm<-confusionMatrix(cmatrix)

>cfm
 Confusion Matrix and Statistics

 Pred
    B  M
  B 85 7
  M 2 25

  Accuracy: 0.9244
  95% CI: (0.8613, 0.9648)
  No Information Rate: 0.7311
  P-Value [Acc > NIR]: 9.846e-08
  Kappa: 0.7977
  Mcnemar's Test P-Value: 0.1824
  Sensitivity: 0.9770
  Specificity: 0.7812
  Pos Pred Value: 0.9239
  NegPredValue: 0.9259
          Prevalence: 0.7311
  Detection Rate: 0.7143
  Detection Prevalence: 0.7731
  Balanced Accuracy: 0.8791

   'Positive' Class: B

 >Accuracy
 [1] "92.4%"
```

Here, the table function stores the information regarding conditional attribute for each attribute and class. The result of table function is stored in the cmatrix variable which is passed in the confusion matrix for the display of the confusion matrix that represents the correct accuracy percentage that is it helps in determining how accurately the model is performing classification.

5. RESULT

Table 4. Result of Accuracy of Naive Bayes Classification

Dataset used: Breast Cancer	Naive Bayes Classification
Kappa	0.7977
Accuracy	92.4%
Positive Class	‘B’

From above Table 4., and also from above execution of code, the result of accuracy percentage for the dataset of breast cancer is 92.4% which is very good for Naive Bayes classification and it has positive class “B” that is, the classified cancer level is Benign(’B’) which is the most probable level in the dataset. Also, the kappa value is 0.7977 which is used to determine how much the classifier classified the values. By using the confusion matrix that is obtained as

```
Pred
   B  M
 B 85 7
 M 2 25
```

Now the graph can be plotted using confusion matrix. Therefore, the confusion matrix is represented by using plot () is given as:

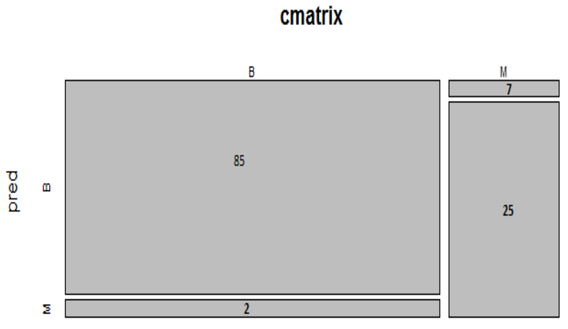

Fig. 4: Confusion matrix of the breast cancer dataset with two levels of class diagnosis: B and M

6. ADVANTAGES AND DISADVANTAGES

6.1 Advantage

The advantages of Naive Bayes are as follows ([16],[17]):

- The Naive Bayes classifier is easy, fast, easy to implement and can handle large datasets in comparison to other classification techniques.
- The Naive Bayes can work on both binary and multi-class classification problems and can handle continuous and discrete datasets.
- As the Naive Bayes relies on the conditional independence assumption. If this independence does not hold, it will still perform better and will give results.
- The Naive Bayes classifier is not sensitive to unrelated features.

6.2 Disadvantage

The disadvantages of Naive Bayes are as follows:

- Naive Bayes is based on the conditional independence assumption but due to this it can lead to accuracy loss.
- For a feature it is required to calculate the likelihood value to predict the value, this aftermath is skewed values towards 0 or 1 probability; thus, results are not good. To overcome this, we need to alter the probabilities to which means it will not be a naive approach.
- For continuous features, it is requirement to use a binning method to convert to disjunctive values, but this needs caution as we may discard a lot of information.

7. CONCLUSION

In this paper, we have introduced the Naive Bayes classification algorithm widely used for handling large datasets. The algorithm is used in spam filtering, recommendation systems etc. The paper also illustrates how the prior probability, posterior probability and likelihood are calculated and then used for classification. We have taken the cancer data set and applied the Naive Bayes classification algorithm in R language to classify cancer either as Malign (M) or Benign (B).

Our study as exhibited in this paper concludes that Naive Bayes classification can give better accuracy for large data sets and is useful in classifying severity of diseases.

ACKNOWLEDGMENT

I would like to express my special thanks to my teacher Dr. Reema Thareja for her valuable support, her patient guidance and for giving me this golden opportunity to work on this research work with her.

REFERENCES

[1] Galit Shmueli, Nitin R. Patel, Peter C. Bruce, 'Classification and Regression Trees', Data Mining for Business Intelligence lecture notes [ebook] (2005) Page-89-90.

[2] Efron B. Mathematics, Bayes' theorem in the 21st century, Science, Published by AAS Vol 340 June (2013); 340:1177-8.

[3] Nikhil Kumar (2017). [online] https://www.geeksforgeeks.org/naive-bayes classifiers/ (Accessed 14 Feburary 2019).

[4] Dalibor Bužić, Jasminka Dobša "Lyrics Classification using Naive Bayes", in MIPRO 2018, 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, At Opatija, Croatia, DOI: 10.23919/MIPRO, (2018).

[5] Sunil Ray (2017) , 6 Easy Steps to Learn Naive Bayes Algorithm with codes in Python and R' [online] September11,https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/ (Accessed 14 February 2019).

[6] Gerardnico (2018) [online] https://gerardnico.com/data_mining/naive_bayes (Accessed 15 February 2019).

[7] S. Sankaranarayanan, T. Pramananda Perumal Analysis of Naive Bayes Classification for Diabetes Mellitus, International Journal of Computer Sciences and Engineering (IJCSE), Vol.-6, Issue-12, Dec (2018), E-ISSN: 2347-2693.

[8] Murphy KP. (2012) Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learningseries). 1st ed. London: The MIT Press; (2012):1.[online] https://doc.lagout.org/science/Artificial%20Intelligence/Machine%20learning/Machine%20Learning_%20A%20Probabilistic%20Perspective%20%5BMurphy%202012-08-24%5D.pdf.

[9] Igor Kononenko "Machine learning for medical diagnosis: history, state of the art and perspective", Artificial Intelligence in Medicine 23(1):89-109 · September (2001) DOI: 10.1016/S0933-3657(01)00077-X.

[10] UC Business Analytics R Programming Guide (2016) [online],uc-r.github.io/naive_bayes (Accessed 18 February 2019).

[11] Sona Taheri, Musa Mammadov, Learning the Naive Bayes Classifier with Optimization Models, International Journal of Applied Mathematics and Computer Science, Vol. 23, No. 4, (2013), 787–795 DOI: 10.2478/amcs-2013-0059.

[12] Wikipedia (2018) [online] https://en.wikipedia.org/wiki/Naive_Bayes_classifier (Accessed 14 February 2019).

[13] Rashmi Jain (2017) [online] https://www.hackerearth.com/blog/machine-learning/introduction-naive-bayes-algorithm-codes-python-r/ February 2 (Accessed 14 February 2019).

[14] Scikit Learn (2013) [online] https://scikit-learn.org/stable/modules/naive_bayes.html#multinomial-naive-bayes (Accessed 20 February 2019).

[15] Zhongheng Zhang "Naive Bayes classification in R", Ann Trans Med; 4(12):241Annals of Translational Medicine, Vol 4, 12 June (2016), DOI: 10.21037/atm.2016.03.38.

[16] Ahmad Ashari, Iman Paryudi, A Min Tjoa Performance Comparison between Naïve Bayes, Decision Tree and k-Nearest Neighbor in Searching Alternative Design in an Energy Simulation Tool, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 4, No. 11, (2013).

[17] D.Sheela Jeyarani, G.Anushya, R.Raja Rajeswari, A.Pethalakshmi , A Comparative Study of Decision Tree and Naive Bayesian Classifiers on Medical Datasets, International Journal of Computer Applications (0975 – 8887) International Conference on Computing and information Technology (IC2IT), 5-7, (2013).

[18] T. John Peter, K. Somasundaram, Study and development of novel feature subsets selection framework for hard disease prediction, International Journal of Scientific and Research Publications, Volume 2, Issue 10, October (2012).

[19] Mai Shouman, Tim Turner, Rob Stocker, Integrating Naive and clustering with a different initial centroid selection methods in the diagnosis of heart disease prediction, in International Conference of Data Mining & Knowledge Management Process (CKDP, (2012)), Dubai, UAE, Published, CS IT CSCP 2012; 125-137.

[20] Ayman M Mansour, Texture Classification using Naïve Bayes Classifier, IJCSNS International Journal of Computer Science and Network Security, VOL.18 No.1, January (2018).

[21] S Vijayarani , S Deepa, Naïve Bayes Classification for Predicting Diseases in Haemoglobin Protein Sequences, International Journal of Computational Intelligence and Informatics, Vol. 3: No. 4, January - March (2014).

[22] GalitShmueli, Nitin R. Patel, Peter C. Bruce**.** Data Mining for Business Intelligence: Concepts, Techniques, and Applications in Microsoft Office Excel with XLMiner.

[23] Tomas Pranckevicius, Virginijus Marcinkevičius , Comparison of Naive Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression Classifiers for Text Reviews Classification, Baltic J. Modern Computing, Vol. 5 (2017), No. 2, 221-232.