Naïve Bayes

Dr. Ami Gates

Reference: Kumar et. Al 2005,2015

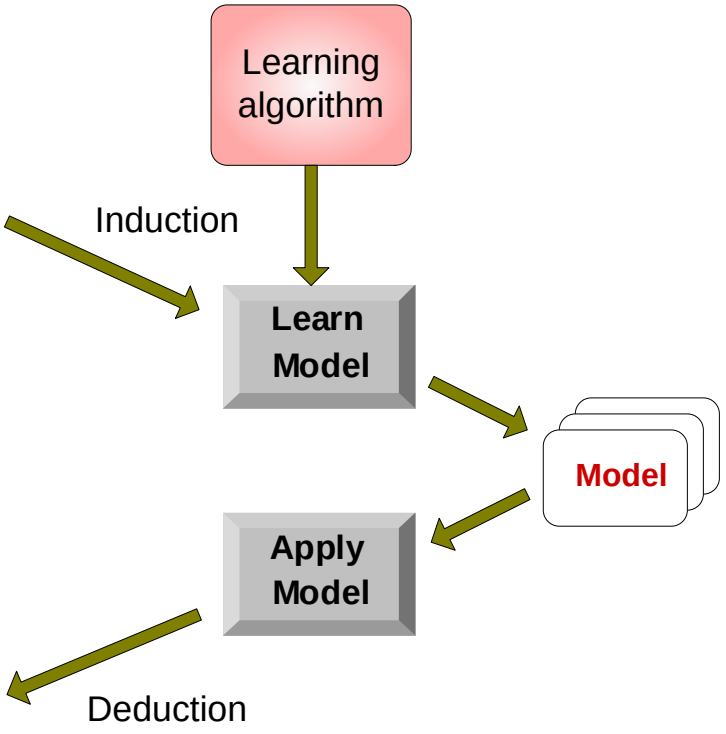
Reminder: A Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

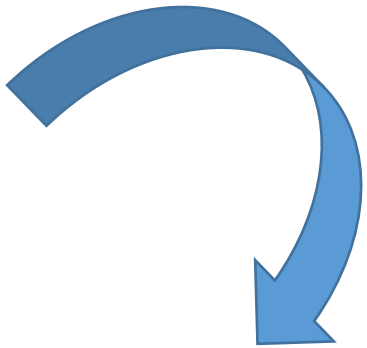
Test Set



Naïve Bayes:

$$P(Y \mid X) = \frac{P(X,Y)}{P(X)}$$

$$P(X \mid Y) = \frac{P(X,Y)}{P(Y)}$$



$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)}$$

Bayes Theorem

- Assume that **attributes are independent**

Let **c** be any **class** or label, Let **x** be a **data vector**

Bayes:

P(c|**x**) = P(**x**|c) P(c) / P(**x**)

= P(x1|c) * P(x2|c) *…*P(xn|c)*P(c) / P(x1) * P(x2) *…*P(xn)

Why**? Because we assume independence.**

P(c|**x**) is the **posterior probability** of the class given the predictor attribute vector **x**

P(**x**|c) is the **Likelihood** – the prob of the attribute vector given the class.

P(c) is the **Class Prior Probability** – the prob of the class

P(**x**) is the **Predictor Prior Probability** – the prob of the attribute vector

Reminder of Conditional Prob Rules and Basic Bayes: FYI

- A probabilistic framework for solving classification problems
- **Conditional Probability**:

$$P(C \mid A) = \frac{P(A,C)}{P(A)}$$

- **Bayes theorem**:

$$P(A \mid C) = \frac{P(A,C)}{P(C)}$$

$$P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)}$$

Example 1: Basic Bayes

Given:

- A doctor knows that meningitis (M) causes stiff neck, S, is 50%. **P(S|M)=.5**
- Prior probability of any patient having meningitis is 1/50000. **P(M)** = **1/50000**
- Prior probability of any patient having stiff neck is 1/20: **P(S) = 1/20**

Question: If a patient has stiff neck, what's the probability he/she has meningitis?

$$P(M \mid S) = \frac{P(S \mid M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Bayesian Classifiers Overview

- Consider each attribute and class label as random and independent variables
- Given a record (data row) with attributes (A_1, A_2,…,A_n)
  - Goal is to predict class C
  - Specifically, we want to find the value of C that **maximizes** P(C| A_1, A_2,…,A_n)
- Can we estimate P(C| A_1, A_2,…,A_n) directly from data?

Bayesian Classifiers

Approach:

- Compute the posterior probability

$P(C \mid A_1, A_2, \ldots, A_n)$ for all values of C using Bayes

$$P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}$$

- Choose value of C that maximizes

$P(C \mid A_1, A_2, \ldots, A_n)$

- **Equivalent to choosing value of C that maximizes**

$P(A_1, A_2, \ldots, A_n \mid C)\, P(C)$

- How to estimate $P(A_1, A_2, \ldots, A_n \mid C)$?

Naïve Bayes Classifier

- **Assume independence** among attributes A_i when class is given:

  $P(A_1, A_2, \ldots, A_n \mid C_j) =$
  $P(A_1 \mid C_j)\, P(A_2 \mid C_j) \ldots P(A_n \mid C_j)$

  Can estimate $P(A_i \mid C_j)$ for all A_i and C_j.

  New point is classified to C_j if
  $P(A_i \mid C_j) * P(C_j)$ is **maximal**.

Example of Naïve Bayes Classifier

Given a Test Record X, classify as Evade E =Yes or No:

Let record X = {Refund R=No, Married M=Yes, Income I =120K}

P(X|E=No) = P(R=No|E=No) * P(M=Yes| E=No) * P(I=120|E=No)

= 4/7 * 4/7 * 0.0072 = .0024 (see slides 12&13)

P(X|E=Yes) = P(R=No|E=Yes) * P(M=Yes| E=Yes) * P(I=120|E=Yes)

= 1 * 0 * 0.0 = 0.0

Since P(X|E=No)P(E=No) > P(X|E=Yes)P(E=Yes)

Therefore P(E=No|X) > P(E=Yes|X)

→ **Class = No**

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Estimating Discrete Prob from Data

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Class: $P(C) = N_c/N$

  P(No) = 7/10

  P(Yes) = 3/10

- For **discrete** attributes:

  $P(A_i|C_k) = |A_{ik}|/ N_c$ (k)

  - where $|A_{ik}|$ is number of instances having attribute A_i and belongs to class C_k

  Examples:

  P(Married=Yes|E=No) = 4/7

  P(Refund=Yes|E=Yes) = 0

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Normal distribution:

$$P(X_i \mid Y_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(X_i - \mu_{ij})^2}{2\sigma_{ij}^2}}$$

  - One for each (X_i,Y_i) pair

- For (Income, Class=No):
  - If Class=No
    - sample mean = 110
    - sample variance = 2975

$$P(Income = 120 \mid No) = \frac{1}{\sqrt{2\pi}(54.54)} e^{-\frac{(120-110)^2}{2(2975)}} = 0.0072$$

The Normal Prob Dist

- The Normal Prob Distribution
- mu is the mean and sigma is the std dev

$$P(A_i \mid c_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(A_i - \mu_{ij})^2}{2\sigma_{ij}^2}}$$

How to Estimate Probabilities from Data?

- **For continuous attributes**:
  - **Discretize** the range into bins
    - one ordinal attribute per bin
  - **Two-way split:** $(A < v)$ or $(A > v)$
    - choose only one of the two splits as new attribute
  - **Probability density estimation:**
    - Assume attribute follows a **normal distribution**
    - **Use data to estimate parameters of distribution** (e.g., mean and standard deviation)
    - Once probability distribution is known, can use it to estimate the conditional probability $P(A_i|c)$

Naïve Bayes Classifier

- If one of the conditional probability is zero, then the entire expression becomes zero
- Probability estimation:

$$\text{Original}: P(A_i \mid C) = \frac{N_{ic}}{N_c}$$

$$\text{Laplace}: P(A_i \mid C) = \frac{N_{ic} + 1}{N_c + c}$$

$$\text{m - estimate}: P(A_i \mid C) = \frac{N_{ic} + mp}{N_c + m}$$

c: number of classes

p: prior probability

m: parameter

Example: Naïve Bayes Classifier

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

A: attributes

M: mammals

N: non-mammals

$$P(A \mid M) = \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06$$

$$P(A \mid N) = \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0042$$

$$P(A \mid M)P(M) = 0.06 \times \frac{7}{20} = 0.021$$

$$P(A \mid N)P(N) = 0.004 \times \frac{13}{20} = 0.0027$$

P(A|M)P(M) > P(A|N)P(N)

→ **Mammals**

Naïve Bayes Summary

- Robust to isolated noise points
- Handle missing values by ignoring the instance during probability estimate calculations
- Robust to irrelevant attributes
- Independence assumption may not hold for some attributes
  - Use other techniques such as Bayesian Belief Networks (BBN)

Naïve Bayes in R

Gates

Link to the Code:

The follow is a specific example of Naïve Bayes in R.

https://drive.google.com/drive/folders/1rXm4jTHMTTjFvHfJ3daCCWNmOdF9tNPl?usp=sharing

You can find my code and data in the above link.

The following slides show examples, but it is best to read about these options and then write your own code.

Libraries: at least these

```
## LIBRARIES
library(stringr)
#install.packages("e1071")
library(e1071)
#install.packages("mlr")
library(mlr)
# install.packages("caret")
library(caret)
#install.packages("naivebayes")
library(naivebayes)
#install.packages("e1071")
library(e1071)
#install.packages("mlr")
library(mlr)
# install.packages("caret")
library(caret)
#install.packages("naivebayes")
library(naivebayes)
library(mclust)
library(cluster)
library(tm)
```

Naïve Bayes

Create a Test Set

Create a Training Set

*** My example to the right is a specific example for my Student App dataset.

Alterations/changes will depend on the data you use.

```
## For Naive Bayes, we will work with the Cleaned
(head(CleanStudentDF,n=10))
## First, create a trainng and testing set and gi
## DO NOT change the CleanStudentDF as you may ne
## Always use copies or create new DFs
## To create the Testing Set, I will use very 7 v
## The Training Set will be all the remaining val
(every7_indexes<-seq(1,nrow(CleanStudentDF),7))
NB_DF_Test=CleanStudentDF[every7_indexes, ]
NB_DF_Train=CleanStudentDF[-every7_indexes, ]
## View the created Test and Train sets
(head(NB_DF_Train,n=10))
(head(NB_DF_Test,n=10))
## Notice that there are still some NAs in TOEFL
## We will have to deal with these....

## Naive Bayes works on numerical data ONLY
## Remove labels and nominal variables, etc.
NB_DF_Train_onlynums <- NB_DF_Train[-c(1,2,3)]
NB_DF_Test_onlynums <- NB_DF_Test[-c(1,2,3,8)]  #
## and we do not want labels in the Test set
NB_Student_TrainLABELS <-NB_DF_Train$Decision
NB_Student_TestLABELS <- NB_DF_Test$Decision
```

Issues with Naïve Bayes

1) NA values or blanks. You can na.pass. This means you lose those rows in the analysis.
2) You can try to fill in the values with a measure, such as mean or median.
3) Naïve Bayes can *only* run on numerical data – so you must remove all nominal/qual/categorical data first.

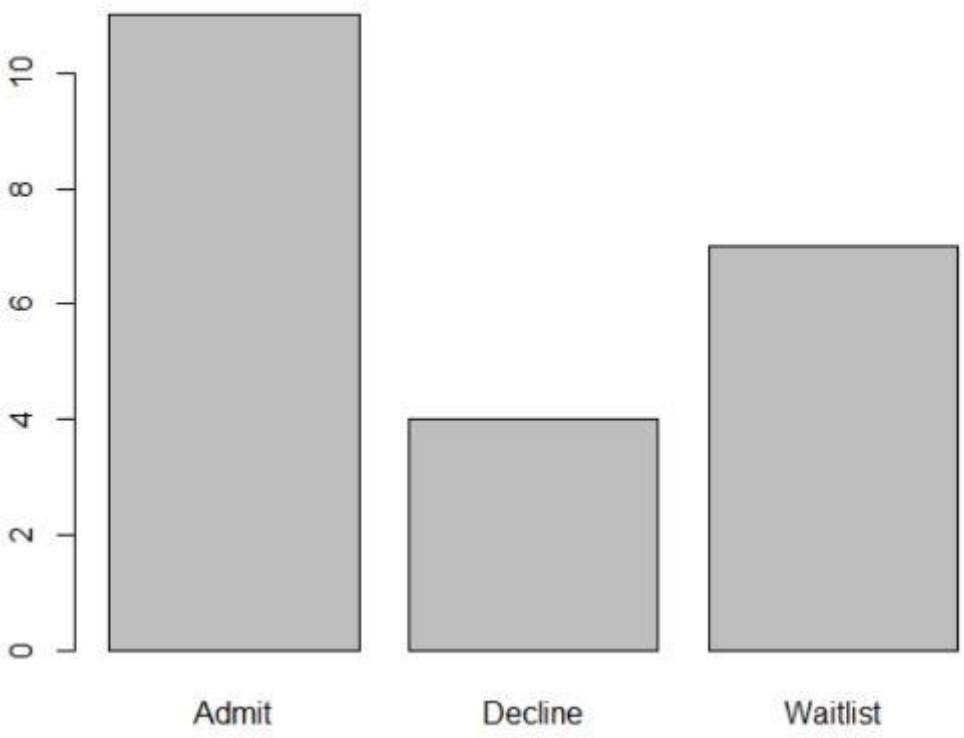
Just an example: Bayes in R

```
## Now we will run the Naive Bayes (NB) classifier. We can do this two ways
## The first way will retain the TOEFL column and will skip (pass) the NAs
## The next way will not use the TOEFL column at all and so will have more data
## but one less variable.

## WAY 1 NB
NBStudentclassfier <- naiveBayes(Decision ~.,data=NB_DF_Train_onlynums, na.action = na.pass)
NBStudentClassifier_Prediction <- predict(NBStudentclassfier, NB_DF_Test_onlynums)
NBStudentclassfier
print(NBStudentClassifier_Prediction)
table(NBStudentClassifier_Prediction,NB_DF_Test$Decision)
plot(NBStudentClassifier_Prediction)

## WAY 2 NB
(head(NB_DF_Train_onlynums, n=5))
(head(NB_DF_Test_onlynums, n=5))
NB_DF_Train_onlynums_noTOEFL <- NB_DF_Train_onlynums[-4]
NB_DF_Test_onlynums_noTOEFL <- NB_DF_Test_onlynums[-4]
NBStudentclassfier2 <- naiveBayes(Decision ~.,data=NB_DF_Train_onlynums_noTOEFL, na.action = na.pass)
NBStudentClassifier_Prediction2 <- predict(NBStudentclassfier2, NB_DF_Test_onlynums_noTOEFL)
NBStudentclassfier2
print(NBStudentClassifier_Prediction2)
table(NBStudentClassifier_Prediction2,NB_DF_Test$Decision)
plot(NBStudentClassifier_Prediction2)
```

Results from the classifier

```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
    Admit   Decline  Waitlist
0.4803150 0.2283465 0.2913386

Conditional probabilities:
          GPA
Y                [,1]       [,2]
  Admit    3.743770 0.13309350
  Decline  3.227241 0.32076801
  Waitlist 3.484865 0.06576987

          WorkExp
Y                [,1]     [,2]
  Admit    1.839344 1.678519
  Decline  2.203448 1.946881
  Waitlist 2.075676 1.370402

          MathTest
Y                [,1]      [,2]
  Admit    963.5738 21.720543
  Decline  797.4828 77.628060
  Waitlist 866.0541  3.431439

> print(NBStudentClassifier_Prediction2)
 [1] Admit    Admit    Admit    Admit    Admit    Admit    Admit
 [8] Admit    Admit    Admit    Admit    Decline  Decline  Decline
[15] Decline  Waitlist Waitlist Waitlist Waitlist Waitlist Waitlist
[22] Waitlist
Levels: Admit Decline Waitlist
> table(NBStudentClassifier_Prediction2,NB_DF_Test$Decision)

NBStudentClassifier_Prediction2 Admit Decline Waitlist
                        Admit       9       2        0
                        Decline     0       4        0
                        Waitlist    0       0        7
```

Quick plot for naïve Bayes



Example 2 in R

The following example is taken directly from:

https://www.r-bloggers.com/understanding-naive-bayes-classifier-using-r/

- R supports a package **called 'e1071'** which provides the naive bayes training function. For this demonstration, we will use the classic titanic dataset and find out the cases which naive bayes can identify as survived.
- The **Titanic dataset in R (not Kaggle)** is a table for about 2200 passengers summarised according to four factors – economic status ranging from 1st class, 2nd class, 3rd class and crew; gender which is either male or female; Age category which is either Child or Adult and whether the type of passenger survived. For each combination of Age, Gender, Class and Survived status, the table gives the number of passengers who fall into the combination.

What is Naïve Bayes?

Naive Bayes – a Not so Naive Algorithm

- The reason that Naive Bayes algorithm is called Naive … is because the algorithm makes a very strong assumption about the data having features **independent of each other** while in reality, they may be dependent in some way.
- In other words, it assumes that the presence of one feature in a class is completely unrelated to the presence of all other features. If this assumption of independence holds, Naive Bayes performs extremely well and often better than other models.
- Naive Bayes can also be used with continuous features but is more suited to categorical variables. If all the input features are categorical, Naive Bayes is recommended. However, in case of numeric features, it makes another strong assumption which is that the numerical variable is normally distributed.

https://www.r-bloggers.com/understanding-naive-bayes-classifier-using-r/

Naïve Bayes Example with R Titanic Data Part 1

```
1  #Getting started with Naive Bayes
2  #Install the package
3  #install.packages("e1071")
4  #Loading the library
5  library(e1071)
6  ?naiveBayes #The documentation
7  #Next load the Titanic dataset
8  data("Titanic")
9  #Save into a data frame and view it
10 Titanic_df=as.data.frame(Titanic)
```

Naïve Bayes Example with R Titanic Data Part 2

```
#Creating data from table
repeating_sequence=rep.int(seq_len(nrow(Titanic_df)),
Titanic_df$Freq)

#This will repeat each combination equal to the
#frequency of each combination
#Create the dataset by row repetition created

Titanic_dataset=Titanic_df[repeating_sequence,]

#We no longer need the frequency, drop the feature

Titanic_dataset$Freq=NULL
```

Naïve Bayes Example with R Titanic Data Part 3

```
#Fitting the Naive Bayes model
Naive_Bayes_Model=naiveBayes(Survived ~.,
data=Titanic_dataset)

#What does the model say?
#Print the model summary

Naive_Bayes_Model
```

results

Naive Bayes Classifier for Discrete Predictors

```
Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
      No      Yes
0.676965 0.323035

Conditional probabilities:
     Class
Y           1st          2nd          3rd          Crew
  No    0.08187919   0.11208054   0.35436242   0.45167785
  Yes   0.28551336   0.16596343   0.25035162   0.29817159

     Sex
Y           Male         Female
  No    0.91543624   0.08456376
  Yes   0.51617440   0.48382560

     Age
Y          Child          Adult
  No    0.03489933   0.96510067
  Yes   0.08016878   0.91983122
```

Results Part 2

The model creates the conditional probability for each feature separately. We also have the a-priori probabilities which indicates the distribution of our data. Let's calculate how we perform on the data.

```
#Prediction on the dataset
NB_Predictions=predict(Naive_Bayes_Model,Titanic_data
#Confusion matrix to check accuracy
table(NB_Predictions,Titanic_dataset$Survived)
NB_Predictions      No      Yes
                No      1364    362
                Yes     126     349
```

Conclusions and Can You DO Better? *Always* ask this question.

We have the results! We are able to classify 1364 out of 1490 “No” cases correctly and 349 out of 711 “Yes” cases correctly. This means the ability of Naive Bayes algorithm to predict “No” cases is about 91.5% but it falls down to only 49% of the “Yes” cases resulting in an overall accuracy of 77.8%

Naive Bayes is a parametric algorithm which implies that you cannot perform differently in different runs as long as the data remains the same.

We will, however, **learn another implementation of Naive Bayes algorithm using the ‘mlr’ package.** Assuming the same session is going on for the readers, I will install and load the package and start fitting a model

Example 3: Using the mlr package

This full example is taken directly from:

https://www.r-bloggers.com/understanding-naive-bayes-classifier-using-r/

Author info is here too as well as all of the code that they used.

Step 1: using mlr

```
#Getting started with Naive Bayes in mlr
#Install the package
#install.packages("mlr")
#Loading the library
library(mlr)
```

The mlr package consists of a lot of models and works by creating tasks and learners which are then trained. Let's create a classification task using the titanic dataset and fit a model with the naive bayes algorithm.

```
#Create a classification task for learning on Titanic
task = makeClassifTask(data = Titanic_dataset, target
#Initialize the Naive Bayes classifier
selected_model = makeLearner("classif.naiveBayes")
#Train the model
NB_mlr = train(selected_model, task)
```

The summary of the model which was printed in e3071 package is stored in learner model. Let's print it and compare

```
#Read the model learned
NB_mlr$learner.model
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
      No      Yes
0.676965 0.323035

Conditional probabilities:
     Class
Y             1st         2nd         3rd          (
    No    0.08187919  0.11208054  0.35436242  0.45:
    Yes   0.28551336  0.16596343  0.25035162  0.29:

     Sex
Y             Male        Female
     No    0.91543624  0.08456376
    Yes    0.51617440  0.48382560

     Age
Y          Child       Adult
    No    0.03489933  0.96510067
    Yes   0.08016878  0.91983122
```

Table of results

The a-priori probabilities and the conditional probabilities for the model are similar to the one calculated by e3071 package as was expected. This means that our predictions will also be the same.

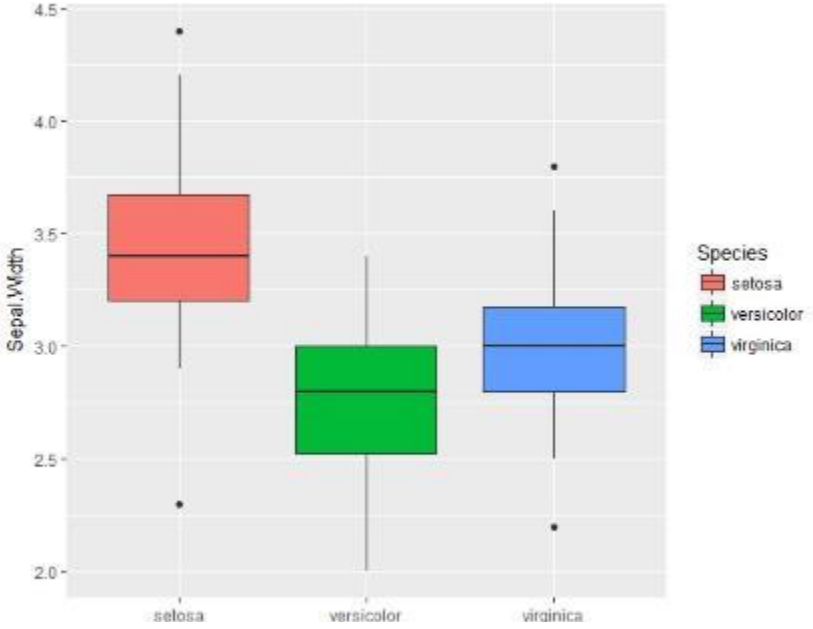
```
#Predict on the dataset without passing the target fe
predictions_mlr = as.data.frame(predict(NB_mlr, newda

##Confusion matrix to check accuracy
table(predictions_mlr[,1],Titanic_dataset$Survived)
        No      Yes
  No    1364    362
  Yes   126     349
```

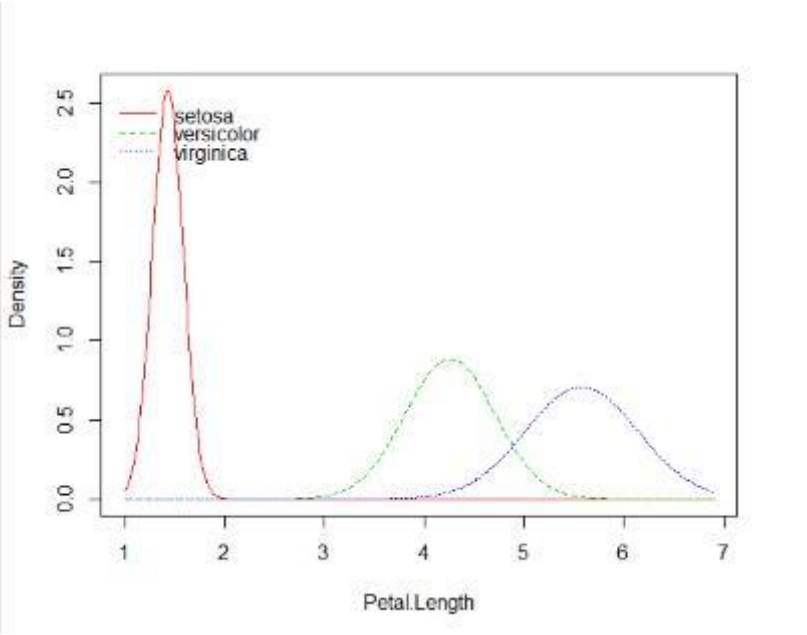
R: Example 4

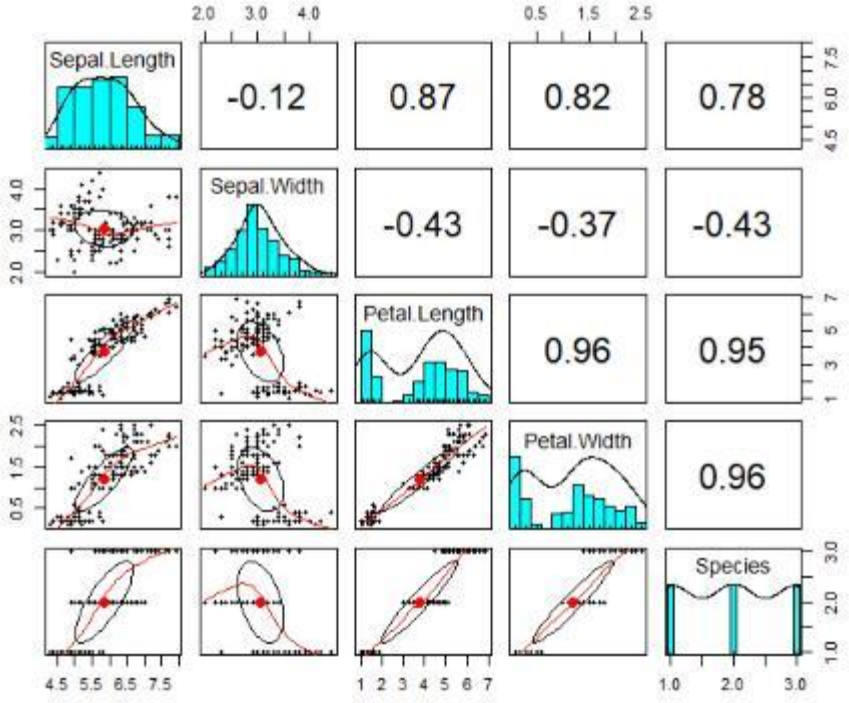
```
## libraries
library(naivebayes)
library(psych)
library(dplyr)
library(ggplot2)

## data
data(iris)
str(iris)
# Species is the label
table(iris$Species)
(head(iris))
(summary(iris))

MyIrisDF <- iris

## Create Train and Test Set
(n <- round(nrow(MyIrisDF)/5))
(s <- sample(1:nrow(MyIrisDF), n))

## The test set is the sample
IrisTest<- MyIrisDF[s,]
## The trainng set is the not sample
IrisTrain <- MyIrisDF[-s,]
## Have a look...
(head(IrisTest,n=5))
(head(IrisTrain,n=5))
```

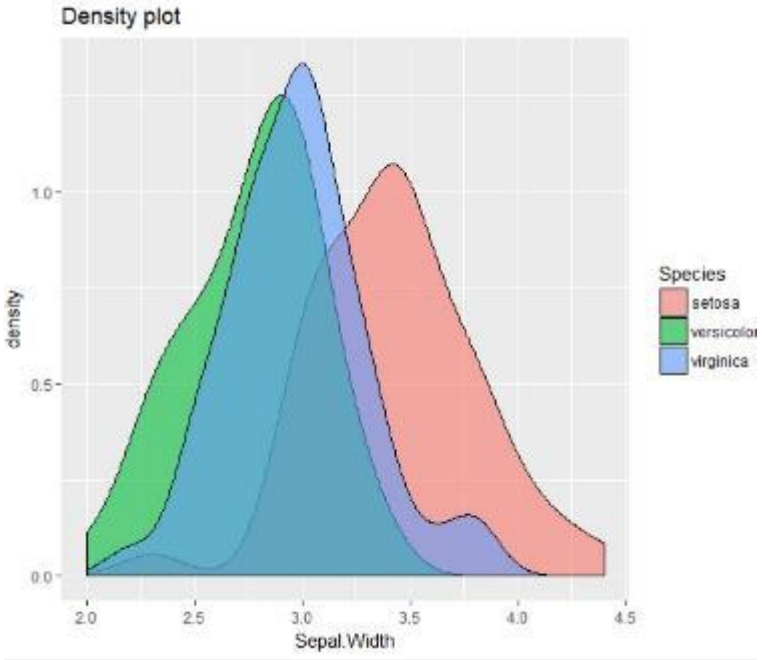
Visual EDA

```
pairs.panels(iris)

iris %>%
  ggplot(aes(x=Species, y=Sepal.Width, fill=Species)) +
  geom_boxplot() +
  ggtitle("Box plot")

iris %>%
  ggplot(aes(x=Sepal.Width, fill=Species)) +
  geom_density(alpha=.6, color="black") +
  ggtitle("Density plot")

plot(iris)
plot(iris$Petal.Length,iris$Petal.Width,col=iris$Species)
```


Sepal.Width
4.5
4.0
3.5
3.0
2.5
2.0
setosa
versicolor
virginica
Species
setosa
versicolor
virginica


Density
2.5
2.0
1.5
1.0
0.5
0.0
setosa
versicolor
virginica
1
2
3
4
5
6
7
Petal.Length


Sepal.Length
-0.12
0.87
0.82
0.78
Sepal.Width
-0.43
-0.37
-0.43
Petal.Length
0.96
0.95
Petal.Width
0.96
Species


Density plot
density
1.0
0.5
0.0
2.0
2.5
3.0
3.5
4.0
4.5
Sepal.Width
Species
setosa
versicolor
virginica

Naïve Bayes

```
##  Naive Bayes
NBModel <- naive_bayes(IrisTrain$Species ~., data=IrisTrain)
print(NBModel)

plot(NBModel)

## predict
NB_p <- predict(NBModel, IrisTest, type="prob")
(head(round(NB_p)))

## classify/predict
NB_p2 <- predict(NBModel, IrisTest, type="class")
(Pred_TABLE <-table(NB_p2, IrisTest$Species))
## accuracy
(sum(diag(Pred_TABLE)/sum(Pred_TABLE)))
```

Suggestions and references

1) Kumar 2005, 2015
2) As with all new things – create a small toy example and do it by hand and with R to see that you get the same results.
3) https://www.r-bloggers.com/understanding-naive-bayes-classifier-using-r/
4) There is always more! Each time you use a model or technique, you will always find new ideas, new improvements, new packages, and new methods to visualize.
5) Ask the question – Can I make this better?