Lab 6: Machine Learning – Naïve Bayes

|  |  |
| --- | --- |
| Due: | Monday, June 24th by the end of class |
| Assignment Type: | Group (Up to 5 – may be randomly assigned) |
| Assignment Title: | Introduction to Machine Learning – Naïve Bayes |
| Style: | One problem at a time on Discord – WORK TOGETHER |
| External Sources: | You are allowed notes, books, and searches |
| Description: | We are going to do assignments dealing with Naïve Bayes. You will be completing this in R. |
| Points | 50 |
| Starting Files | On Blackboard (Download there) |

Enter your group member names here:

You should turn in your lab8\_1.R file, lab8\_2.R file, and your dataset for lab8\_2. Zip all of them and submit them.

This lab was introduced by Jeremy Dixon and based on <https://www.edureka.co/blog/naive-bayes-in-r/>

# Using Naïve Bayes in R

1. You need to download the file **diabetes.csv** for this problem. Download it from Blackboard.

For this problem, we are going to be working through an example of Naïve Bayes to be able to predict new patients. We are going to use machine learning to predict whether a person has diabetes or not.

You will need to create a new project in R and add the diabetes.csv file to that new folder. In addition, you will need to update the code below to use your path instead of mine. You can check your progress at the bottom of this page: <https://www.edureka.co/blog/naive-bayes-in-r/>

The data in diabetes.csv is as follows:

The given data set contains 100s of observations of patients along with their health details. Here’s a list of the predictor variables that will help us classify a patient as either Diabetic or Normal:

* Pregnancies: Number of pregnancies so far
* Glucose: Plasma glucose concentration
* BloodPressure: Diastolic blood pressure (mm Hg)
* SkinThickness: Triceps skin fold thickness (mm)
* Insulin: 2-Hour serum insulin (mu U/ml)
* BMI: Body mass index (weight in kg/(height in m)^2)
* DiabetesPedigreeFunction: Diabetes pedigree function
* Age: Age (years)

The response variable or the output variable is:

Outcome: Class variable (0 or 1)

For this, we will need the following packages. You should only need to load them once so after you install all of them, you can remove this part.

#Loading required packages

install.packages('tidyverse')

install.packages('ggplot2')

install.packages('caret')

install.packages('caretEnsemble')

install.packages('psych')

install.packages('Amelia')

install.packages('mice')

install.packages('GGally')

install.packages('rpart')

install.packages('randomForest')

install.packages("klaR")

install.packages("haven")

You will need the code for each in the library.

library(tidyverse) #data science library

library(ggplot2) #plotting library

library(caret) #Short for "Classification and Regression Training" used for Bayes

library(caretEnsemble) #makes ensembles of caret models

library(psych) #used by social scientists

library(Amelia) #used to replace missing data

library(mice) #used to replace missing data

library(GGally) #extension for ggplot2

library(rpart) #used to split the data recursively

library(randomForest) #random forest algorithm for classification and regression

library(klaR) #Evaluates the performance of a classification method

#Reading data into R

data<- read.csv("~/UMBC/2020/Summer/CMSC491/R/naive\_bayes/diabetes.csv")

#Setting outcome variables as categorical - identifies the outcome variable from the dataset

data$Outcome <- factor(data$Outcome, levels = c(0,1), labels = c("False", "True"))

#Studying the structure of the data

str(data)

head(data)

describe(data)

#Data Cleaning

#Convert '0' values into NA

data[, 2:7][data[, 2:7] == 0] <- NA

#visualize the missing data

#plots a missingness map showing where missingness occurs in the dataset

#from Amelia

missmap(data)

#Uses mice package to predict missing values

mice\_mod <- mice(data[, c("Glucose","BloodPressure","SkinThickness","Insulin","BMI")], method='rf')

mice\_complete <- complete(mice\_mod)

#Transfer the predicted missing values into the main data set

data$Glucose <- mice\_complete$Glucose

data$BloodPressure <- mice\_complete$BloodPressure

data$SkinThickness <- mice\_complete$SkinThickness

data$Insulin<- mice\_complete$Insulin

data$BMI <- mice\_complete$BMI

#visualize the missing data again to double check

missmap(data)

#Data Visualization

#Visual 1

#Shows the age distribution count by outcome (diabetic or not diabetic)

ggplot(data, aes(Age, colour = Outcome)) +

geom\_freqpoly(binwidth = 1) + labs(title="Age Distribution by Outcome")

#visual 2

#Shows the pregnancies by outcome

c <- ggplot(data, aes(x=Pregnancies, fill=Outcome, color=Outcome)) +

geom\_histogram(binwidth = 1) + labs(title="Pregnancy Distribution by Outcome")

c + theme\_bw()

#visual 3

P <- ggplot(data, aes(x=BMI, fill=Outcome, color=Outcome)) +

geom\_histogram(binwidth = 1) + labs(title="BMI Distribution by Outcome")

P + theme\_bw()

#visual 4

ggplot(data, aes(Glucose, colour = Outcome)) +

geom\_freqpoly(binwidth = 1) + labs(title="Glucose Distribution by Outcome")

#visual 5

#Produces a matrix of scatter plots for visualizing the correlation between variables

#From GGally

ggpairs(data)

## IF YOU GET WARNINGS – just keep going. Question 2 is to try and debug those warnings.

#Building a model

#split data into training and test data sets

#Creates a series of test/training partitions from Carat

indxTrain <- createDataPartition(y = data$Outcome,p = 0.75,list = FALSE)

training <- data[indxTrain,]

testing <- data[-indxTrain,] #Check dimensions of the split > prop.table(table(data$Outcome)) \* 100

#create objects x which holds the predictor variables and y which holds the response variables

x = training[,-9]

y = training$Outcome

#Functions for Naive Bayes (provided by R)

library(e1071)

#Sets up a grid of tuning parameters for a number of classification and regression routines

#From Carat

model = train(x,y,'nb',trControl=trainControl(method='cv',number=10))

#Model Evaluation

#Predict testing set

Predict <- predict(model,newdata = testing ) #Get the confusion matrix to see accuracy value and other parameter values > confusionMatrix(Predict, testing$Outcome )

#Confusion Matrix and Statistics

#Plot Variable performance

#Tracks the changes in model statistics from Carat

X <- varImp(model)

plot(X)

Paste your screenshot here. Add your lab8\_1.R file to your submission.

# Fixing Warnings for Naïve Bayes in R

1. 1a. From problem 1, you may get warnings. Try to figure out what the warnings are and why they exist. Can you remove them? If you are able to fix these warnings, write a comment in your lab8\_1.R file describing in detail what you did to fix it. If you are not able, it is ok (we won’t grade this question)

# Using Naïve Bayes in R

1. For this problem, you are going to use a dataset that you find to repeat the process above. You can use any dataset you like – just Google “Naïve bayes dataset”. The diabetes dataset above is from Kaggle – they have more data available. Try to do at least 2 meaningful visualizations, fix any missing data, and using the training/predicting features described above.

Add your dataset file to your lab8 submission and your lab8\_2.R file.