Assignment 3

Yashwanth k

2023-10-10

#summary

1. If an accident has just been reported and there is no additional information available, it should be assumed that there may be injuries (INJURY = Yes). This is so that the maximum degree of injury from the accident is indicated by the variable MAX_SEV_IR. If MAX_SEV_IR equals 1 or 2, it is assumed that there is some degree of injury (INJURY = Yes), according to the instructions. No injury is implied (INJURY = No) if MAX_SEV_IR is equal to 0. Therefore, it is wise to assume that there may be some degree of injury in the absence of more information until the contrary is established by new information.

- There are total of " 20721 NO and yes are 21462"

2. Created a pivot table considering only Three variables Injury, Weather, Traffic, using First 24 rows so that we can get a new data frame with 24 records and only 3 vairables and also dropped the Injury because we are going to calculate bayes probability

- we're using the probabilities calculated in Step 1 to classify the accidents. For each accident in the first 24 records, we calculate the probability of an injury and classify it based on the cutoff of 0.5. The classifications can be printed.

3. while the majority of classifications between the exact and Naive Bayes methods are equivalent, there are discrepancies in the "Yes" category, where a few records are classified differently. These differences may be due to the simplifying assumption of independence made by the Naive Bayes classifier, which may not hold for all cases. When exact probabilities and conditional dependencies are crucial, the exact Bayes method is more appropriate, but it can be computationally intensive for larger datasets.

- The overall error rate on the validation set is approximately 0.47 when expressed as a decimal. It indicates that the Naive Bayes classifier performs very well on this dataset, with a high level of accuracy.

  Confusion Matrix and Statistics -

```
Accuracy    :   0.5
Sensitivity : 0.15635
Specificity : 0.87083
```

PROBLEM STATEMENT

The file accidentsFull.csv contains information on 42,183 actual automobile accidents in 2001 in the United States that involved one of three levels of injury: NO INJURY, INJURY, or FATALITY. For each accident, additional information is recorded, such as day of week, weather conditions, and road type. A firm might be interested in developing a system for quickly classifying the severity of an accident based on initial reports and associated data in the system (some of which rely on GPS-assisted reporting).

Our goal here is to predict whether an accident just reported will involve an injury (MAX_SEV_IR = 1 or 2) or will not (MAX_SEV_IR = 0). For this purpose, create a dummy variable called INJURY that takes the value "yes" if MAX_SEV_IR = 1 or 2, and otherwise "no."

1. Using the information in this dataset, if an accident has just been reported and no further information is available, what should the prediction be? (INJURY = Yes or No?) Why?
2. Select the first 24 records in the dataset and look only at the response (INJURY) and the two predictors WEATHER_R and TRAF_CON_R. Create a pivot table that examines INJURY as a function of the two predictors for these 24 records. Use all three variables in the pivot table as rows/columns.

1.Compute the exact Bayes conditional probabilities of an injury (INJURY = Yes) given the six possible combinations of the predictors.

2.Classify the 24 accidents using these probabilities and a cutoff of 0.5.

3.Compute manually the naive Bayes conditional probability of an injury given WEATHER_R = 1 and TRAF_CON_R = 1.

4.Run a naive Bayes classifier on the 24 records and two predictors. Check the model output to obtain probabilities and classifications for all 24 records. Compare this to the exact Bayes classification. Are the resulting classifications equivalent? Is the ranking (= ordering) of observations equivalent?

3. Let us now return to the entire dataset. Partition the data into training (60%) and validation (40%).

` 1.Run a naive Bayes classifier on the complete training set with the relevant predictors (and INJURY as the response). Note that all predictors are categorical. Show the confusion matrix. 2.What is the overall error of the validation set?

```
#Loading the libraries that are required for the task
library(class)
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(e1071)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#Loading the data set and assigning it to Accidents variable.
Accidents <- read.csv("accidentsFull.csv")
dim(Accidents)
```

```
## [1] 42183    24
```

```
Accidents$INJURY = ifelse(Accidents$MAX_SEV_IR %in% c(1,2),"yes","no")
table(Accidents$INJURY) # as yes is greater than no
```

```
##
##    no   yes
## 20721 21462
```

```
t(t(names(Accidents)))
```

```
##       [,1]
##  [1,] "HOUR_I_R"
##  [2,] "ALCHL_I"
##  [3,] "ALIGN_I"
##  [4,] "STRATUM_R"
##  [5,] "WRK_ZONE"
##  [6,] "WKDY_I_R"
##  [7,] "INT_HWY"
##  [8,] "LGTCON_I_R"
##  [9,] "MANCOL_I_R"
## [10,] "PED_ACC_R"
## [11,] "RELJCT_I_R"
## [12,] "REL_RWY_R"
## [13,] "PROFIL_I_R"
## [14,] "SPD_LIM"
## [15,] "SUR_COND"
## [16,] "TRAF_CON_R"
## [17,] "TRAF_WAY"
## [18,] "VEH_INVL"
## [19,] "WEATHER_R"
## [20,] "INJURY_CRASH"
## [21,] "NO_INJ_I"
## [22,] "PRPTYDMG_CRASH"
## [23,] "FATALITIES"
## [24,] "MAX_SEV_IR"
## [25,] "INJURY"
```

1.Using the information in this dataset, if an accident has just been reported and no further information is available, what should the prediction be? (INJURY = Yes or No?) Why?

```
# Find the majority class and print it
majority_class <- names(sort(table(Accidents$INJURY), decreasing = TRUE))[1]
cat("Predicted initial class based on majority:", majority_class)
```

```
## Predicted initial class based on majority: yes
```

2.Select the first 24 records in the dataset and look only at the response (INJURY) and the two predictors WEATHER_R and TRAF_CON_R. Create a pivot table that examines INJURY as a function of the two predictors for these 12 records. Use all three variables in the pivot table as rows/columns. 1-Compute the exact Bayes conditional probabilities of an injury (INJURY = Yes) given the six possible combinations of the predictors. 2-Classify the 24 accidents using these probabilities and a cutoff of 0.5. 3-Compute manually the naive Bayes conditional probability of an injury given WEATHER_R = 1 and TRAF_CON_R = 1. 4- Run a naive Bayes classifier on the 24 records and two predictors. Check the model output to obtain probabilities and classifications for all 24 records. Compare this to the exact Bayes classification. Are the resulting classifications equivalent? Is the ranking (= ordering) of observations equivalent?

```
#Creating the pivot tables
sub_Accidents <- Accidents[1:24,c("INJURY","WEATHER_R","TRAF_CON_R")]
sub_Accidents
```

```
##    INJURY WEATHER_R TRAF_CON_R
## 1     yes         1          0
## 2      no         2          0
## 3      no         2          1
## 4      no         1          1
## 5      no         1          0
## 6     yes         2          0
## 7      no         2          0
## 8     yes         1          0
## 9      no         2          0
## 10     no         2          0
## 11     no         2          0
## 12     no         1          2
## 13    yes         1          0
## 14     no         1          0
## 15    yes         1          0
## 16    yes         1          0
## 17     no         2          0
## 18     no         2          0
## 19     no         2          0
## 20     no         2          0
## 21    yes         1          0
## 22     no         1          0
## 23    yes         2          2
## 24    yes         2          0
```

```
pi_table1 <- ftable(sub_Accidents)


pi_table2 <- ftable(sub_Accidents[,-1])
```

---

pair_1 = sum(sub_Accidents[WEATHER_R == 1 & sub_Accidents]TRAF_CON_R == 0) pair_2 = sum(sub_Accidents[WEATHER_R == 1 & sub_Accidents]TRAF_CON_R == 1) pair_3 = sum(sub_Accidents[WEATHER_R == 1 & sub_Accidents]TRAF_CON_R == 2)
pair_4 = sum(sub_Accidents[WEATHER_R == 2 & sub_Accidents]TRAF_CON_R == 0) pair_5 = sum(sub_Accidents[WEATHER_R == 2 & sub_Accidents]TRAF_CON_R == 1) pair_6 = sum(sub_Accidents[WEATHER_R == 2 & sub_Accidents]TRAF_CON_R == 2)

dual_1 = sum(sub_Accidents[WEATHER_R == 1 & sub_Accidents]TRAF_CON_R == 0 & sub_Accidents$INJURY==“YES”)

dual_2 = sum(sub_Accidents[WEATHER_R == 1 & sub_Accidents]TRAF_CON_R == 1 & sub_Accidents$INJURY==“YES”)

dual_3 = sum(sub_Accidents[WEATHER_R == 1 & sub_Accidents]TRAF_CON_R == 2 & sub_Accidents$INJURY==“YES”)

dual_4 = sum(sub_Accidents[WEATHER_R == 2 & sub_Accidents]TRAF_CON_R == 0 & sub_Accidents$INJURY==“YES”)

dual_5 = sum(sub_Accidents[WEATHER_R == 2 & sub_Accidents]TRAF_CON_R == 1 & sub_Accidents$INJURY==“YES”)

dual_6 = sum(sub_Accidents[WEATHER_R == 2 & sub_Accidents]TRAF_CON_R == 2 & sub_Accidents$INJURY==“YES”)

prob_1 <- dual_1 / pair_1 cat(“P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 0):”, prob_1, “”)

prob_2 <- dual_2 / pair_2 cat(“P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 1):”, prob_2, “”)

prob_3 <- dual_3 / pair_3 cat(“P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 2):”, prob_3, “”)

prob_4 <- dual_4 / pair_4 cat(“P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 0):”, prob_4, “”)

prob_5 <- dual_5 / pair_5 cat(“P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 1):”, prob_5, “”)

prob_6 <- dual_6 / pair_6 cat(“P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 2):”, prob_6, “”)

#2.1

```
#bayes
#INJURY = YES
pair_1 = pi_table1[3,1]/pi_table2[1,1]
cat("P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 0):", pair_1, "\n")
```

```
## P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 0): 0.6666667
```

```
pair_2 = pi_table1[3,2]/pi_table2[1,2]
cat("P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 1):", pair_2, "\n")
```

```
## P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 1): 0
```

```
pair_3 = pi_table1[3,3]/pi_table2[1,3]
cat("P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 2):", pair_3, "\n")
```

```
## P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 2): 0
```

```
pair_4 = pi_table1[4,1]/pi_table2[2,1]
cat("P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 0):", pair_4, "\n")
```

```
## P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 0): 0.1818182
```

```
pair_5 = pi_table1[4,2]/pi_table2[2,2]
cat("P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 1):", pair_5, "\n")
```

```
## P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 1): 0
```

```
pair_6 = pi_table1[4,3]/pi_table2[2,3]
cat("P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 2):", pair_6, "\n")
```

```
## P(INJURY = Yes | WEATHER_R = 2 and TRAF_CON_R = 2): 1
```

```
#Now we check the condition whether Injury = no

dual_1 = pi_table1[1,1]/pi_table2[1,1]
cat("P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 0):", dual_1, "\n")
```

```
## P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 0): 0.3333333
```

```
dual_2 = pi_table1[1,2]/pi_table2[1,2]
cat("P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 1):", dual_2, "\n")
```

```
## P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 1): 1
```

```
dual_3 = pi_table1[1,3]/pi_table2[1,3]
cat("P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 2):", dual_3, "\n")
```

```
## P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 2): 1
```

```
dual_4 = pi_table1[2,1]/pi_table2[2,1]
cat("P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 0):", dual_4, "\n")
```

```
## P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 0): 0.8181818
```

```
dual_5 = pi_table1[2,2]/pi_table2[2,2]
cat("P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 1):", dual_5, "\n")
```

```
## P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 1): 1
```

```
dual_6 = pi_table1[2,3]/pi_table2[2,3]
cat("P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 2):", dual_6, "\n")
```

```
## P(INJURY = no | WEATHER_R = 2 and TRAF_CON_R = 2): 0
```

```
#Now probability of the total occurences.
```

---

else if(sub_Accidents WEATHER_R[i] == 1 && sub_Accidents TRAF_CON_R[i] == 0){ prob_injury[i] = dual_1

} else if(sub_Accidents WEATHER_R[i] == 1 && sub_Accidents TRAF_CON_R[i] == 1){ prob_injury[i] = dual_2 } else if(sub_Accidents WEATHER_R[i] == 1 && sub_Accidents TRAF_CON_R[i] == 2){ prob_injury[i] = dual_3 } else if(sub_Accidents WEATHER_R[i] == 2 && sub_Accidents TRAF_CON_R[i] == 0){ prob_injury[i] = dual_4 } else if(sub_Accidents WEATHER_R[i] == 2 && sub_Accidents TRAF_CON_R[i] == 1){ prob_injury[i] = dual_5 }

```r
#cutoff is 0.5 and for 24 records
# Assuming you have calculated the conditional probabilities already, you can use them to classi
fy the 24 accidents.
# Let's say you have a dataframe named 'new_data' containing these 24 records.

prob_injury <- rep(0,24)
for(i in 1:24){
  print(c(sub_Accidents$WEATHER_R[i],sub_Accidents$TRAF_CON_R[i]))

  if(sub_Accidents$WEATHER_R[i] == "1" && sub_Accidents$TRAF_CON_R[i] == "0"){
    prob_injury[i] = pair_1

  } else if (sub_Accidents$WEATHER_R[i] == "1" && sub_Accidents$TRAF_CON_R[i] == "1"){
    prob_injury[i] = pair_2

  } else if (sub_Accidents$WEATHER_R[i] == "1" && sub_Accidents$TRAF_CON_R[i] == "2"){
    prob_injury[i] = pair_3

  }
  else if (sub_Accidents$WEATHER_R[i] == "2" && sub_Accidents$TRAF_CON_R[i] == "0"){
    prob_injury[i] = pair_4

  } else if (sub_Accidents$WEATHER_R[i] == "2" && sub_Accidents$TRAF_CON_R[i] == "1"){
    prob_injury[i] = pair_5

  }
  else if(sub_Accidents$WEATHER_R[i] == "2" && sub_Accidents$TRAF_CON_R[i] == "2"){
    prob_injury[i] = pair_6
  }

}
```

```
## [1] 1 0
## [1] 2 0
## [1] 2 1
## [1] 1 1
## [1] 1 0
## [1] 2 0
## [1] 2 0
## [1] 1 0
## [1] 2 0
## [1] 2 0
## [1] 2 0
## [1] 1 2
## [1] 1 0
## [1] 1 0
## [1] 1 0
## [1] 1 0
## [1] 2 0
## [1] 2 0
## [1] 2 0
## [1] 2 0
## [1] 1 0
## [1] 1 0
## [1] 2 2
## [1] 2 0
```

```
#cutoff 0.5

sub_Accidents$prob_injury = prob_injury
sub_Accidents$pred.prob  = ifelse(sub_Accidents$prob_injury>0.5, "yes","no")


head(sub_Accidents)
```

```
##   INJURY WEATHER_R TRAF_CON_R prob_injury pred.prob
## 1    yes         1          0   0.6666667       yes
## 2     no         2          0   0.1818182        no
## 3     no         2          1   0.0000000        no
## 4     no         1          1   0.0000000        no
## 5     no         1          0   0.6666667       yes
## 6    yes         2          0   0.1818182        no
```

#Compute manually the naive Bayes conditional probability of an injury given WEATHER_R = 1 and TRAF_CON_R = 1.

```
IY = pi_table1[3,2]/pi_table2[1,2]
I = (IY * pi_table1[3, 2]) / pi_table2[1, 2]
cat("P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 1):", IY, "\n")
```

```
## P(INJURY = Yes | WEATHER_R = 1 and TRAF_CON_R = 1): 0
```

```
IN = pi_table1[1,2]/pi_table2[1,2]
N = (IY * pi_table1[3, 2]) / pi_table2[1, 2]
cat("P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 1):", IN, "\n")
```

```
## P(INJURY = no | WEATHER_R = 1 and TRAF_CON_R = 1): 1
```

#2.4

```
new_b <- naiveBayes(INJURY ~ TRAF_CON_R + WEATHER_R,
                  data = sub_Accidents)

new_Accidents <- predict(new_b, newdata = sub_Accidents,type = "raw")
sub_Accidents$nbpred.prob <- new_Accidents[,2]


new_c <- train(INJURY ~ TRAF_CON_R + WEATHER_R,
      data = sub_Accidents, method = "nb")
```

```
## Warning: model fit failed for Resample02: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R
```

```
## Warning: model fit failed for Resample04: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R
```

```
## Warning: model fit failed for Resample07: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R
```

```
## Warning: model fit failed for Resample08: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R
```

```
## Warning: model fit failed for Resample09: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R
```

```
## Warning: model fit failed for Resample10: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R
```

```
## Warning: model fit failed for Resample13: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R
```

```
## Warning: model fit failed for Resample16: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R
```

```
## Warning: model fit failed for Resample17: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R
```

```
## Warning: model fit failed for Resample21: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R
```

```
## Warning: model fit failed for Resample24: usekernel=FALSE, fL=0, adjust=1 Error in NaiveBaye
s.default(x, y, usekernel = FALSE, fL = param$fL, ...) :
##   Zero variances for at least one class in variables: TRAF_CON_R
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.
```

```
predict(new_c, newdata = sub_Accidents[,c("INJURY", "WEATHER_R", "TRAF_CON_R")])
```

```
##  [1] yes no  no  yes yes no  no  yes no  no  no  yes yes yes yes yes yes no  no  no
## [20] no  yes yes no  no
## Levels: no yes
```

```
predict(new_c, newdata = sub_Accidents[,c("INJURY", "WEATHER_R", "TRAF_CON_R")],
                                    type = "raw")
```

```
##  [1] yes no  no  yes yes no  no  yes no  no  no  yes yes yes yes yes yes no  no  no
## [20] no  yes yes no  no
## Levels: no yes
```

#Let us now return to the entire dataset. Partition the data into training (60%) and validation (40%). Run a naive Bayes classifier on the complete training set with the relevant predictors (and INJURY as the response). Note that all predictors are categorical. Show the confusion matrix.What is the overall error of the validation set?

```
accident = Accidents[c(-24)]

set.seed(1)
acc.index = sample(row.names(accident), 0.6*nrow(accident)[1])
valid.index = setdiff(row.names(accident), acc.index)


acc.df = accident[acc.index,]
valid.df= accident[valid.index,]

dim(acc.df)
```

```
## [1] 25309    24
```

```
dim(valid.df)
```

```
## [1] 16874    24
```

```
norm.values <- preProcess(acc.df[,], method = c("center", "scale"))
acc.norm.df <- predict(norm.values, acc.df[, ])
valid.norm.df <- predict(norm.values, valid.df[, ])

levels(acc.norm.df)
```

```
## NULL
```

```
class(acc.norm.df$INJURY)
```

```
## [1] "character"
```

```
acc.norm.df$INJURY <- as.factor(acc.norm.df$INJURY)

class(acc.norm.df$INJURY)
```

```
## [1] "factor"
```

```
nb_model <- naiveBayes(INJURY ~ WEATHER_R + TRAF_CON_R, data = acc.norm.df)

predictions <- predict(nb_model, newdata = valid.norm.df)

#Ensure that factor levels in validation dataset match those in training dataset
valid.norm.df$INJURY <- factor(valid.norm.df$INJURY, levels = levels(acc.norm.df$INJURY))

# Show the confusion matrix
confusionMatrix(predictions, valid.norm.df$INJURY)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   no  yes
##        no  1285 1118
##        yes 6934 7537
##
##                Accuracy : 0.5228
##                  95% CI : (0.5152, 0.5304)
##     No Information Rate : 0.5129
##     P-Value [Acc > NIR] : 0.005162
##
##                   Kappa : 0.0277
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.15635
##             Specificity : 0.87083
##          Pos Pred Value : 0.53475
##          Neg Pred Value : 0.52083
##              Prevalence : 0.48708
##          Detection Rate : 0.07615
##    Detection Prevalence : 0.14241
##       Balanced Accuracy : 0.51359
##
##        'Positive' Class : no
##
```

```
# Calculate the overall error rate
error_rate <- 1 - sum(predictions == valid.norm.df$INJURY) / nrow(valid.norm.df)
error_rate
```

```
## [1] 0.4771838
```