Naive bayes

Jaymeen Gandhi

4/24/2020

require(tidyverse)

## Loading required package: tidyverse

## -- Attaching packages ----------------------------------------------------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.0 v purrr 0.3.4  
## v tibble 3.0.1 v dplyr 0.8.5  
## v tidyr 1.0.2 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0

## -- Conflicts -------------------------------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

churn = read\_csv("C:/Users/jayme/Downloads/WA\_Fn-UseC\_-Telco-Customer-Churn.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_character(),  
## SeniorCitizen = col\_double(),  
## tenure = col\_double(),  
## MonthlyCharges = col\_double(),  
## TotalCharges = col\_double()  
## )

## See spec(...) for full column specifications.

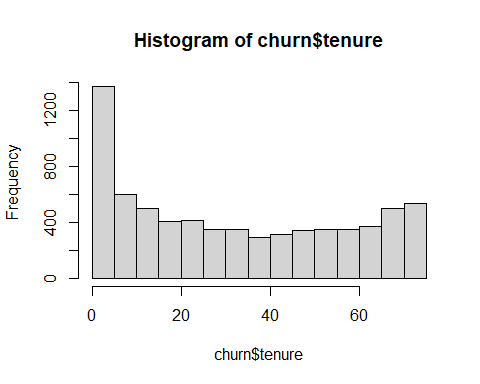
##Preprocessing

churn %>% head(10)

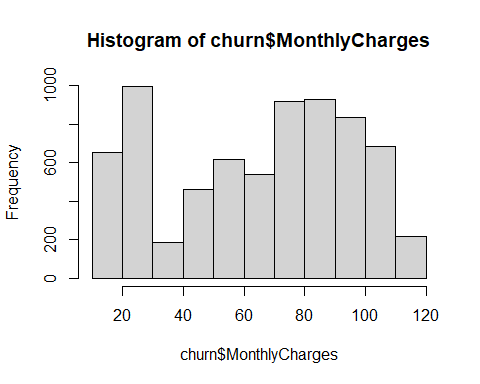
## # A tibble: 10 x 21  
## customerID gender SeniorCitizen Partner Dependents tenure PhoneService  
## <chr> <chr> <dbl> <chr> <chr> <dbl> <chr>   
## 1 7590-VHVEG Female 0 Yes No 1 No   
## 2 5575-GNVDE Male 0 No No 34 Yes   
## 3 3668-QPYBK Male 0 No No 2 Yes   
## 4 7795-CFOCW Male 0 No No 45 No   
## 5 9237-HQITU Female 0 No No 2 Yes   
## 6 9305-CDSKC Female 0 No No 8 Yes   
## 7 1452-KIOVK Male 0 No Yes 22 Yes   
## 8 6713-OKOMC Female 0 No No 10 No   
## 9 7892-POOKP Female 0 Yes No 28 Yes   
## 10 6388-TABGU Male 0 No Yes 62 Yes   
## # ... with 14 more variables: MultipleLines <chr>, InternetService <chr>,  
## # OnlineSecurity <chr>, OnlineBackup <chr>, DeviceProtection <chr>,  
## # TechSupport <chr>, StreamingTV <chr>, StreamingMovies <chr>,  
## # Contract <chr>, PaperlessBilling <chr>, PaymentMethod <chr>,  
## # MonthlyCharges <dbl>, TotalCharges <dbl>, Churn <chr>

##Since the senior citizen column is in numerical form we need to convert it to categorical variable.  
  
churn = churn %>% mutate(SeniorCitizen=as.character(SeniorCitizen))  
  
churn = churn %>% select(-customerID)

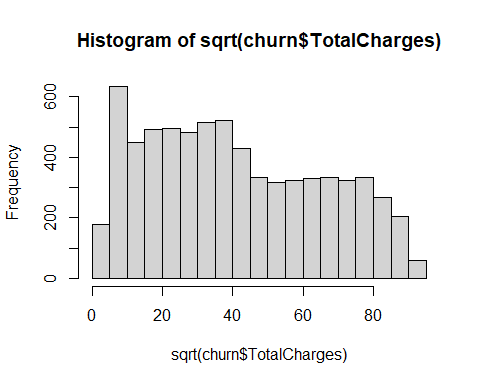
## Now we have to check whether all the numeric variable are normally distributed or not.  
## Check it with the help of histograms.  
  
hist(churn$tenure) ##Not normally distribued requires binning



hist(churn$MonthlyCharges) ##Not normally distribued requires binning



hist(sqrt(churn$TotalCharges)) ## Taking the square root works over here so we do not need binning.



##Binning the numeric data  
##churn%>%mutate(tenure=cut(tenure,2,labels = c("low","high"))) # for knowledge  
churn2 = churn%>%mutate(tenure=cut(tenure,2),MonthlyCharges=cut(MonthlyCharges,4),TotalCharges=sqrt(TotalCharges))  
churn2 %>% head(10)

## # A tibble: 10 x 20  
## gender SeniorCitizen Partner Dependents tenure PhoneService MultipleLines  
## <chr> <chr> <chr> <chr> <fct> <chr> <chr>   
## 1 Female 0 Yes No (-0.0~ No No phone ser~  
## 2 Male 0 No No (-0.0~ Yes No   
## 3 Male 0 No No (-0.0~ Yes No   
## 4 Male 0 No No (36,7~ No No phone ser~  
## 5 Female 0 No No (-0.0~ Yes No   
## 6 Female 0 No No (-0.0~ Yes Yes   
## 7 Male 0 No Yes (-0.0~ Yes Yes   
## 8 Female 0 No No (-0.0~ No No phone ser~  
## 9 Female 0 Yes No (-0.0~ Yes Yes   
## 10 Male 0 No Yes (36,7~ Yes No   
## # ... with 13 more variables: InternetService <chr>, OnlineSecurity <chr>,  
## # OnlineBackup <chr>, DeviceProtection <chr>, TechSupport <chr>,  
## # StreamingTV <chr>, StreamingMovies <chr>, Contract <chr>,  
## # PaperlessBilling <chr>, PaymentMethod <chr>, MonthlyCharges <fct>,  
## # TotalCharges <dbl>, Churn <chr>

set.seed(1234)  
churn2 = churn2 %>% mutate(id = row\_number())  
training\_data = churn2 %>% sample\_frac(0.7)  
testing\_data = anti\_join(churn2 , training\_data , by ='id')

require(e1071)

## Loading required package: e1071

require(caret)

## Loading required package: caret

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

nb\_model = naiveBayes(training\_data[,-20],as.factor(training\_data$Churn),laplace = 0.1)  
prediction = predict(nb\_model,testing\_data[,-20])  
  
confusionMatrix(as.factor(prediction),as.factor(testing\_data$Churn))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 1094 108  
## Yes 453 458  
##   
## Accuracy : 0.7345   
## 95% CI : (0.7151, 0.7532)  
## No Information Rate : 0.7321   
## P-Value [Acc > NIR] : 0.4139   
##   
## Kappa : 0.4327   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.7072   
## Specificity : 0.8092   
## Pos Pred Value : 0.9101   
## Neg Pred Value : 0.5027   
## Prevalence : 0.7321   
## Detection Rate : 0.5177   
## Detection Prevalence : 0.5689   
## Balanced Accuracy : 0.7582   
##   
## 'Positive' Class : No   
##

## Inorder to improve the model we can add laplace to the model(values between 0.1 to 1) or we can play with the cuts.

##Now our predictions are in the Yes or no form so we can make our predictions to our raw format.  
  
raw\_predictions = predict(nb\_model,testing\_data[,-20], type = "raw")  
#raw\_predictions

## We can improve the accuracy by using the probablities that is 0.3.

new\_prediction= ifelse(raw\_predictions[,2]>.3, 'Yes','No')  
confusionMatrix(as.factor(new\_prediction), as.factor(testing\_data$Churn))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 1031 90  
## Yes 516 476  
##   
## Accuracy : 0.7132   
## 95% CI : (0.6934, 0.7324)  
## No Information Rate : 0.7321   
## P-Value [Acc > NIR] : 0.9761   
##   
## Kappa : 0.4097   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.6665   
## Specificity : 0.8410   
## Pos Pred Value : 0.9197   
## Neg Pred Value : 0.4798   
## Prevalence : 0.7321   
## Detection Rate : 0.4879   
## Detection Prevalence : 0.5305   
## Balanced Accuracy : 0.7537   
##   
## 'Positive' Class : No   
##