Naive-Bayes

Using the library “e1071”, I ran the function naiveBayes on the data and used the test data to make the predictions.

```
##Libraries-----------------------------------------
library(tree)
library(ISLR)
library(boot)
library(xgboost)
library(tidyverse)

## ── Attaching packages
─────────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.1.0     ✔ purrr   0.3.0
## ✔ tibble  2.0.1     ✔ dplyr   0.7.8
## ✔ tidyr   0.8.2     ✔ stringr 1.4.0
## ✔ readr   1.3.1     ✔ forcats 0.3.0

## ── Conflicts ──────────────────────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ✖ dplyr::slice()  masks xgboost::slice()

library(leaflet)
library(stringr)
library(rgdal)

## Loading required package: sp

## rgdal: version: 1.4-3, (SVN revision 828)
##  Geospatial Data Abstraction Library extensions to R successfully loaded
##  Loaded GDAL runtime: GDAL 2.1.3, released 2017/20/01
##  Path to GDAL shared files:
/Library/Frameworks/R.framework/Versions/3.5/Resources/library/rgdal/gdal
##  GDAL binary built with GEOS: FALSE
##  Loaded PROJ.4 runtime: Rel. 4.9.3, 15 August 2016, [PJ_VERSION: 493]
##  Path to PROJ.4 shared files:
/Library/Frameworks/R.framework/Versions/3.5/Resources/library/rgdal/proj
##  Linking to sp version: 1.3-1

library(lubridate)

##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(forecast)
library(DT)
library(prophet)
```

```
## Loading required package: Rcpp
```

```
## Loading required package: rlang
```

```
##
## Attaching package: 'rlang'
```

```
## The following objects are masked from 'package:purrr':
##
##     %@%, as_function, flatten, flatten_chr, flatten_dbl,
##     flatten_int, flatten_lgl, flatten_raw, invoke, list_along,
##     modify, prepend, splice
```

```
library(caret)
```

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

```
##
## Attaching package: 'lattice'
```

```
## The following object is masked from 'package:boot':
##
##     melanoma
```

```
##
## Attaching package: 'caret'
```

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

```
library(fastDummies)
library(caret)
library(class)
library(e1071)
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

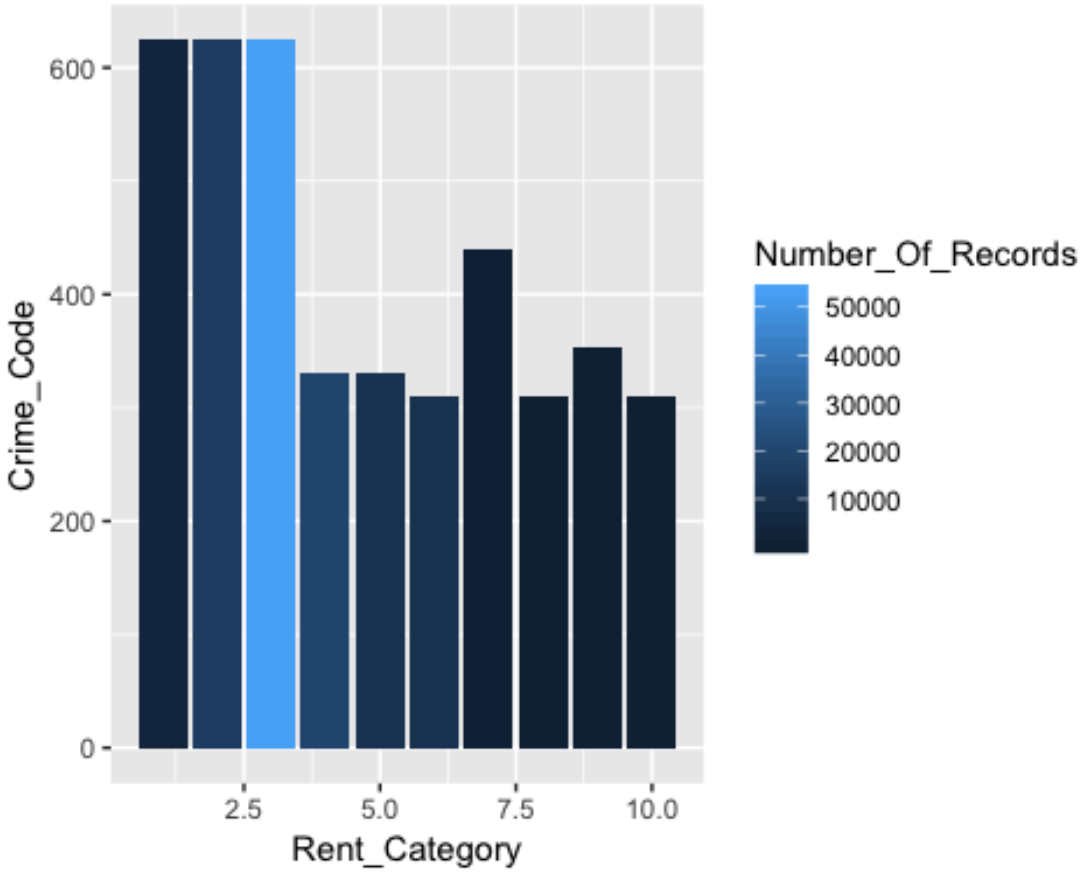
```r
library(ggplot2)
library(lattice)


##Read the data
mergedf <-
read.csv("~/Desktop/GitAdd/Data_Mining/Files/Outputs/CrimeRentData.csv")


##Remove columns
mergedf <- within(mergedf, rm("X"))
mergedf <- within(mergedf, rm("AreaName"))
mergedf <- within(mergedf, rm("CrimeCodeDescription"))
mergedf <- within(mergedf, rm("DRNumber"))
mergedf <- within(mergedf, rm("Location.x"))
mergedf <- within(mergedf, rm("PremiseDescription"))
mergedf <- within(mergedf, rm("Variable"))
mergedf <- within(mergedf, rm("Location.y"))
mergedf <- within(mergedf, rm("Date"))
mergedf <- within(mergedf, rm("VictimDescent"))
#mergedf <-within(mergedf,rm("Neighborhood"))


mergedf = na.omit(mergedf)
mergedf$Year <- as.factor(mergedf$Year)
mergedf$CrimeCode <- as.factor(mergedf$CrimeCode)
mergedf$Neighborhood <- as.factor(mergedf$Neighborhood)

##Remove extra columns
mergedf<-within(mergedf, rm("DateOccurred"))
mergedf<-within(mergedf, rm("Tract"))
mergedf<-within(mergedf, rm("ReportingDistrict"))
mergedf<-within(mergedf, rm("TimeOccured"))
mergedf<-within(mergedf, rm("VictimAge"))
mergedf<-within(mergedf, rm("VictimSex"))
mergedf<-within(mergedf, rm("PremiseCode"))


##Create Amount categories
mergedf$Amount <- as.numeric(mergedf$Amount)

ra <- range(mergedf$Amount)
div <- (ra[2]-ra[1])/10
ini <- ra[1]
br <- rep(0,11)
br[1]<-ra[1]
for(i in 2:11){
  ini<-ini+div
  br[i]<-ini
```

```
}

mergedf$Renth <- cut(mergedf$Amount,
                     breaks=br,
                     labels=c("1","2","3","4","5","6","7","8","9","10"))

mergedf <- within(mergedf,rm("Amount"))

#Split the data
set.seed(12345)
inTrain <- createDataPartition(mergedf$Renth, p=0.7, list=FALSE)
dftrain <- data.frame(mergedf[inTrain,])
dftest <- data.frame(mergedf[-inTrain,])

##Regression
fit1<- naiveBayes(Renth~., data=mergedf)



##Confusion Matrix
ActualData <- dftest$Renth
PredictedData<- predict(fit1,newdata=dftest[,-5])
(table1 <- table(ActualData,PredictedData))

##            PredictedData
## ActualData      1      2      3      4      5      6      7      8      9
##          1    3305    440   1054     50    669     36      0      0      0
##          2    1313  12710  18519    639    840    286      6     40      0
##          3     158   5648 121163   7907    491   1088      8     22      2
##          4     998    820  21748  21937   3746   3373    209     43     47
##          5    1888    191   3968   7221   7570   3924    412     63     43
##          6     396    147   4143   5773   2528  11457    321     14      4
##          7       0      0     27     42    148      9    511     67      4
##          8       0      0     23      0      0      5     49    129      0
##          9       0      0      1     32     25     16     31      0     31
##          10      0      0      0      9     75     16     35      5      3
##            PredictedData
## ActualData     10
##          1       0
##          2       0
##          3       0
##          4      70
##          5      53
##          6       1
##          7       6
##          8       6
##          9       2
##          10     43
```

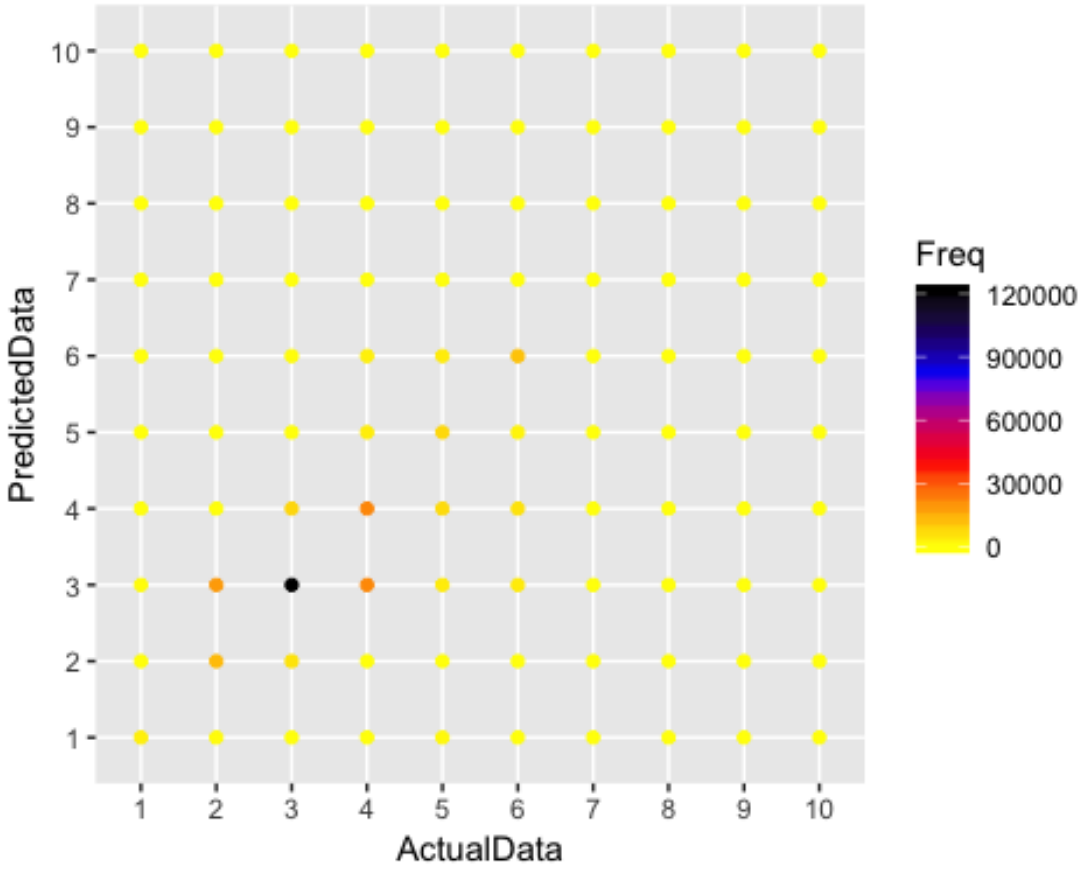
```r
##Accuracy
sum=0
er<-rep(0,10)
for(i in 1:10){
  er[i]<-table1[i,i]
  sum=sum+table1[i,i]
}

(acc= sum/nrow(dftest))
```

```
## [1] 0.6367203
```

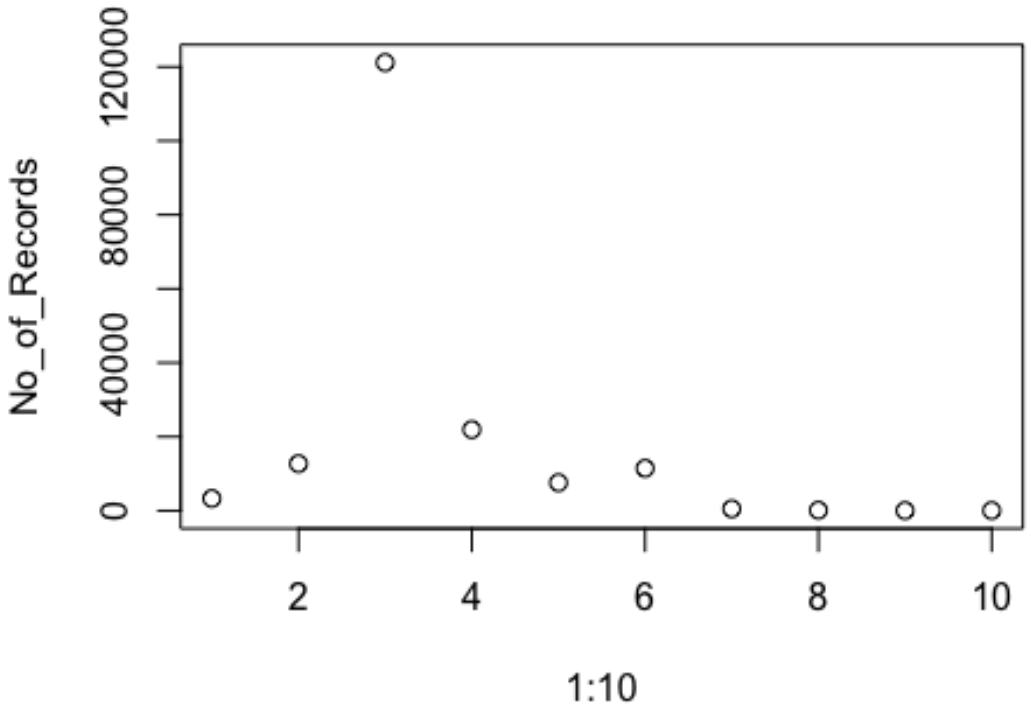
```r
##Graph

##Graph for max crime type for each rent category
Rent_Category<- c(1,2,3,4,5,6,7,8,9,10)
Crime_Code<- c(624,624,624,330,330,310,440,310,354,310)
Number_Of_Records <- c(3385,15032,53317,19151,9764,9958,343,101,72,94)
d <- data.frame(Rent_Category,Crime_Code,Number_Of_Records)

#visualize training set
a<-ggplot(d,aes(x=Rent_Category,y=Crime_Code,fill=Number_Of_Records))+
  geom_bar(stat="identity")
a
```




```
##Graph for visual representation of confusion matrix

cbp1 <- c("yellow","red","blue","black")
table2 <- data.frame(table1)
a<- ggplot(table2, aes(ActualData, PredictedData,color=Freq)) +
  geom_point()

a+scale_color_gradientn(colours=cbp1)
```




```
No_of_Records<-rep(0,10)
for(i in 1:10){
  No_of_Records[i]<- table1[i,i]
}

plot(1:10,No_of_Records)
```




```
#write.csv(counts, "~/Desktop/DMProject/NBGraph1.csv")
```