Final Project

Matthew Michael Collins

5/1/2022

Models Utilized

Data mining with Naiive Bayes

Naiive Bayes: Reading in Data

```
# Loading package
library(e1071)
library(caTools)
library(caret)
```

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

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

```
set.seed(100)
train<-read.csv(file = 'training_final.csv', header=TRUE)
test<-read.csv(file = 'test_final.csv', header=TRUE)

train[,13]<-as.factor(train[,13])
y_test<-as.factor(train[,13])
x_test<-test
```

Selection

I chose sig1, sig2, sig7 and sig8 becuase the data when plotted shows two kinds distribution: unifrom and right skewed.

Data preprocesing

This function removes all the NA rows from the dataset. This function also finds the outlying data and removes it from the set.

```r
for (i in which(sapply(train, is.numeric))) {
  train[is.na(train[, i]), i] <- mean(train[, i],  na.rm = TRUE)
}

#install.packages("outliers")
library(outliers)

outlier_tf = outlier(train$sig1,logical=TRUE)

#What were the outliers
find_outlier = which(outlier_tf==TRUE,arr.ind=TRUE)

sum(outlier_tf)
```

```
## [1] 83
```

```r
train = train[-find_outlier,]
nrow(train)
```

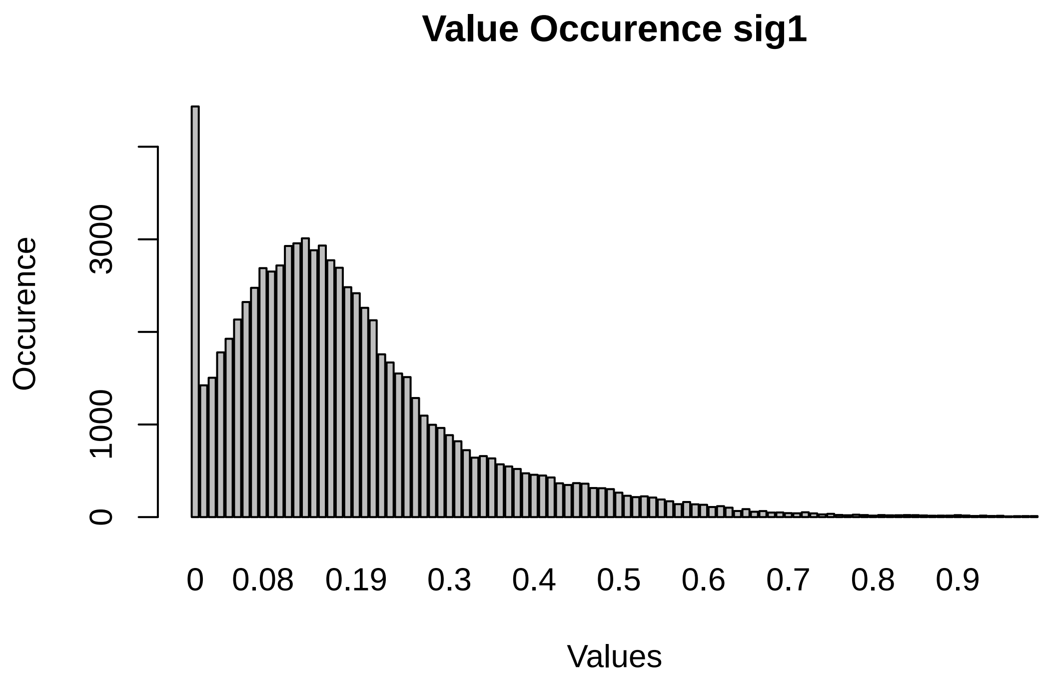
```
## [1] 79963
```

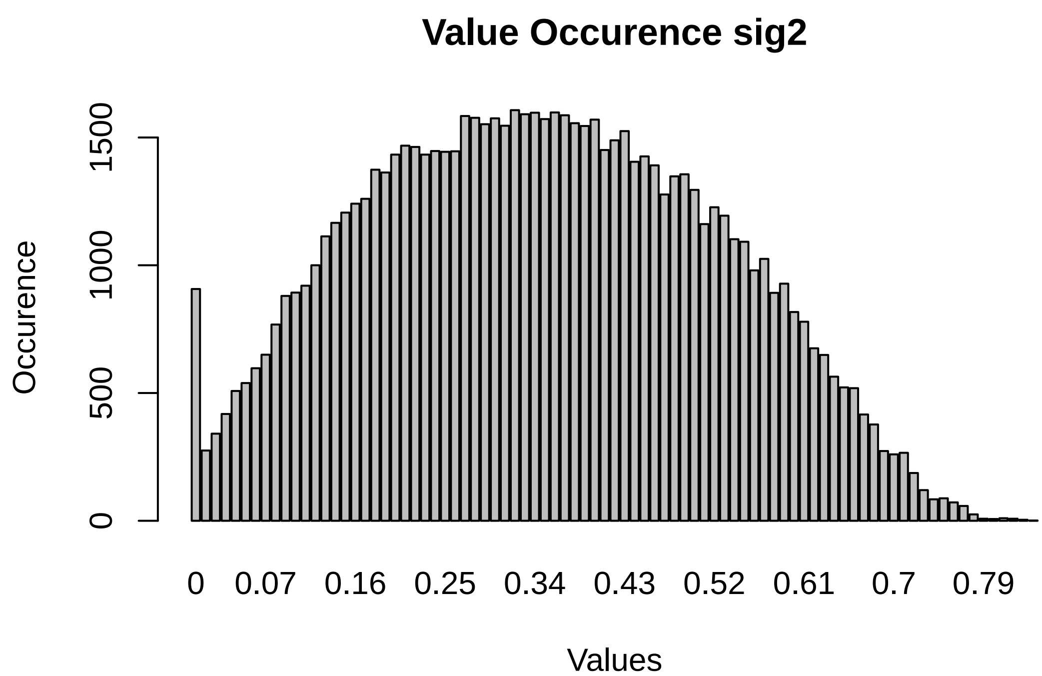
Data Transformation

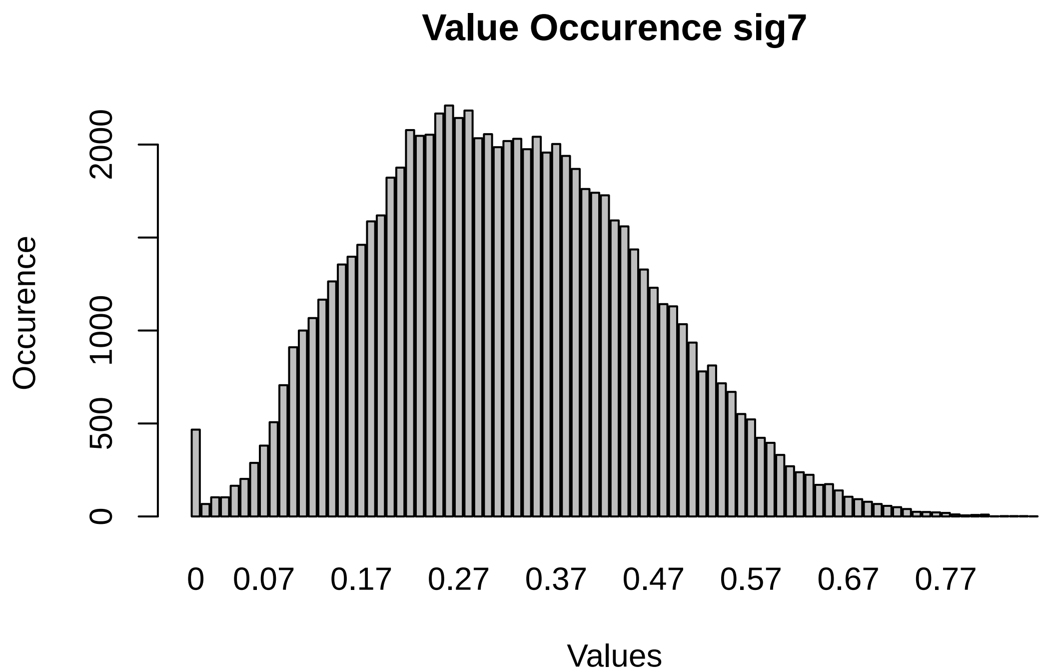
Generate a random sample of "data_set_size" indexes and then Assign the data to a new training set

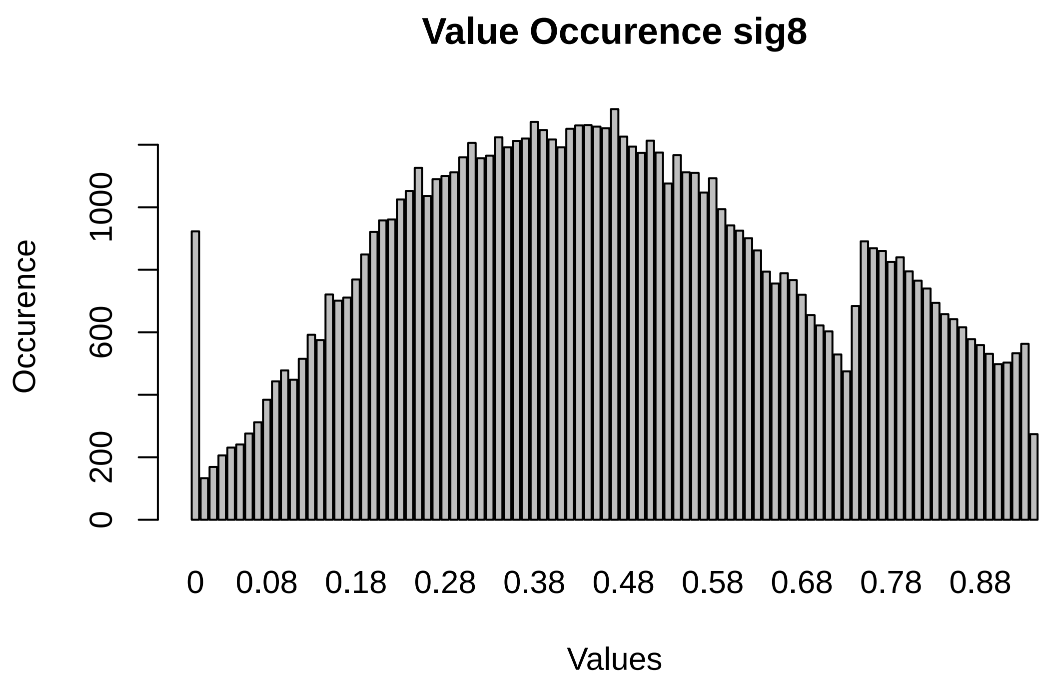
```r
new_train = data.frame(sig1 = train$sig1,
                       sig2 = train$sig2,
                       sig7= train$sig7,
                       sig8= train$sig8)
data_set_size <- round(nrow(new_train)/2)
indexes <- sample(1:nrow(new_train), size = data_set_size)
train2 <- new_train[indexes,]
```

Plots of sig1, sig2, sig7 and sig8









Note that each of these values is uniform distribution or right skewed data.

Misclasification of the Train Data

```
fit<-naiveBayes(relevance~.,data=train)
answer<-predict(fit,train2)
sum(train[,13]!=answer)/length(train[,13])
```

```
## [1] 0.4809599
```

Success Rate Train Data

```
mean(answer==train[,13])
```

```
## [1] 0.5190401
```

Final Misclasification Against Test Data

```
answer2<-predict(fit,x_test)
sum(train[,13]!=answer2)/length(train[,13])
```

```
## [1] 0.4465815
```

Final Success Rate Against Test Data

```
mean(answer2==train[,13])
```

```
## [1] 0.5534185
```

4. Data Mining

I chose Naiive Bayes because it is a simple technique for constructing classifiers. Bayes classifiers also treat each value of a particular feature as independent of the value of any other feature.

Interpretation/Evaluation Misclassification Error on Training

Using sig2 through sig8 because they are both uniform distribution

```
fit2<-naiveBayes(relevance~.,data=train)
answer<-predict(fit2,train2[,2:4])
sum(train[,13]!=answer)/length(train[,13])
```

```
## [1] 0.483186
```

Misclassification With sig1

```r
fit3<-naiveBayes(relevance~.,data=train)
answer<-predict(fit3,train2[,1])
sum(train[,13]!=answer)/length(train[,13])
```

```
## [1] 0.4371022
```

Final Thoughts

Using data with similar distribution results in better classification with the model.

Write to .txt

```r
write(answer2, file="answer.txt",ncol=1)
```