

EDA Project

Rahul Rajput 19BCE1084 Garima Rout 19BCE1144

20/04/2022

R Markdown

```
rm(list=ls())
df<-read.csv("EDA_PROJECT.csv")
df
```

##	Name	Sentence
## 1	Rahul	you have to do it fast rather preapare for circumstances
## 2	Rahul	I got the full marks hurrah
## 3	Rahul	You made a mistake go and clear it
## 4	Rahul	I lost my chance to pass this
## 5	Rahul	She is very happy after the successful interview
## 6	Rahul	Please don't leave me I already lost others
## 7	Rahul	ok I will do this on time
## 8	Rahul	Sorry dad I didn't pass in math exam
## 9	Rahul	I'm feeling pretty good right now
## 10	Nakul	Are you coming out tonight
## 11	Nakul	I would like some coffee please
## 12	Nakul	You wait till morning
## 13	Nakul	Don't talk be quiet
## 14	Garima	You can go now
## 15	Garima	finally,I got it
## 16	Chandani	She likes the gift.
## 17	Chandani	I asked him one simple question and he bit my head off.
## 18	Chandani	After he failed his English exam, he was depressed for a week.
## 19	Chandani	Call me any day.
## 20	Chandani	I'll ruin your career.
## 21	Chandani	You are on top of the world when you feel wonderful.
## 22	Chandani	I'm sorry to hear that.
## 23	Chandani	She went out of the room in anger.
## 24	Chandani	I was in seventh heaven when I landed my dream job.
## 25	Aditi	ok fine will do something
## 26	Aditi	I am smiling because of that.
## 27	Aditi	I failed to do so.
## 28	Aditi	Who has lost their mind
## 29	Aditi	I am resting.
## 30	Aditi	His father is dead for a month now.
## 31	Aditi	I am excited for the event
## 32	Aditi	The man shouted at his staff
## 33	Aditya	The earth goes round the earth.
## 34	Aditya	He is feeling relieved after the session.
## 35	Aditya	Rakhi was crying since she lost her puppy.

```
## 36 Aditya He would be travelling to Chennai.
## 37 Aditya It devastates me to hear of their loss.
## 38 Aditya Rahul is thrilled to know about his promotion
## 39 Aditya I will sue you for your anti-social actions.
## 40 Aditya Do not dare talk to an elder in that way.
## Mood
## 1 Angry
## 2 Happy
## 3 Angry
## 4 Sad
## 5 Happy
## 6 Sad
## 7 Normal
## 8 Sad
## 9 Happy
## 10 Normal
## 11 Normal
## 12 Angry
## 13 Angry
## 14 Normal
## 15 Happy
## 16 Happy
## 17 Angry
## 18 Sad
## 19 Normal
## 20 Sad
## 21 Happy
## 22 Sad
## 23 Angry
## 24 Happy
## 25 Normal
## 26 Happy
## 27 Sad
## 28 Angry
## 29 Normal
## 30 Sad
## 31 Happy
## 32 Angry
## 33 Normal
## 34 Happy
## 35 Sad
## 36 Normal
## 37 Sad
## 38 Happy
## 39 Angry
## 40 Angry
```

```
library(tm)
```

```
## Loading required package: NLP
```

```
corpus<-Corpus(VectorSource(df$Sentance))
#inspect(corpus[1:3])
corpus<-tm_map(corpus,tolower)
```

```
## Warning in tm_map.SimpleCorpus(corpus, tolower): transformation drops documents
corpus<-tm_map(corpus,removeNumbers)

## Warning in tm_map.SimpleCorpus(corpus, removeNumbers): transformation drops
## documents
corpus<-tm_map(corpus,removePunctuation)

## Warning in tm_map.SimpleCorpus(corpus, removePunctuation): transformation drops
## documents
corpus<-tm_map(corpus,removeWords,c(stopwords('english'), "and", "are", "the",
                                     "both", "appears", "within",
                                     "others", "clear", "seen",

## Warning in tm_map.SimpleCorpus(corpus, removeWords, c(stopwords("english"), :
## transformation drops documents
corpus<-tm_map(corpus,stripWhitespace)

## Warning in tm_map.SimpleCorpus(corpus, stripWhitespace): transformation drops
## documents
inspect(corpus)

## <<SimpleCorpus>>
## Metadata: corpus specific: 1, document level (indexed): 0
## Content: documents: 40
##
## [1] fast rather preapare circumstances got full marks hurrah
## [3] made mistake go lost chance pass
## [5] happy successful interview please dont leave already lost
## [7] ok will time sorry dad didnt pass math exam
## [9] im feeling pretty good right now coming tonight
## [11] like coffee please wait till morning
## [13] dont talk quiet can go now
## [15] finallyi got likes gift
## [17] asked one simple question bit head failed english exam depressed week
## [19] call day ill ruin career
## [21] top world feel wonderful im sorry hear
## [23] went room anger seventh heaven landed dream job
## [25] ok fine will something smiling
## [27] failed lost mind
## [29] resting father dead month now
## [31] excited event man shouted staff
## [33] earth goes round earth feeling relieved session
## [35] rakhi crying since lost puppy travelling chennai
## [37] devastates hear loss rahul thrilled know promotion
## [39] will sue antisocial actions dare talk elder way

tdm<-DocumentTermMatrix(corpus)

inspect(tdm)

## <<DocumentTermMatrix (documents: 40, terms: 109)>>
## Non-/sparse entries: 126/4234
## Sparsity : 97%
## Maximal term length: 13
```

```
## Weighting      : term frequency (tf)
## Sample        :
## Terms
## Docs dont exam feeling got lost now pass please sorry will
## 1      0    0      0  0    0  0    0      0    0    0
## 17     0    0      0  0    0  0    0      0    0    0
## 18     0    1      0  0    0  0    0      0    0    0
## 2      0    0      0  1    0  0    0      0    0    0
## 21     0    0      0  0    0  0    0      0    0    0
## 24     0    0      0  0    0  0    0      0    0    0
## 35     0    0      0  0    1  0    0      0    0    0
## 6      1    0      0  0    1  0    0      1    0    0
## 8      0    1      0  0    0  0    1      0    1    0
## 9      0    0      1  0    0  1    0      0    0    0
```

```
tdm_sparse <- removeSparseTerms(tdm,0.97)
tdm_dm1 <- as.data.frame(as.matrix(tdm_sparse)) # count matrix

tdm_df1 <- as.matrix((tdm_dm1 > 0) + 0) # binary instance matrix

tdm_df1 <- as.data.frame(tdm_df1)

tdm_df1 <- cbind(tdm_df1,df$Mood)

tdm_df1
```

```
##      got lost pass dont please will exam sorry feeling now talk failed hear
## 1      0    0    0    0      0    0    0    0      0  0    0      0    0
## 2      1    0    0    0      0    0    0    0      0  0    0      0    0
## 3      0    0    0    0      0    0    0    0      0  0    0      0    0
## 4      0    1    1    0      0    0    0    0      0  0    0      0    0
## 5      0    0    0    0      0    0    0    0      0  0    0      0    0
## 6      0    1    0    1      1    0    0    0      0  0    0      0    0
## 7      0    0    0    0      0    1    0    0      0  0    0      0    0
## 8      0    0    1    0      0    0    1    1      0  0    0      0    0
## 9      0    0    0    0      0    0    0    0      1  1    0      0    0
## 10     0    0    0    0      0    0    0    0      0  0    0      0    0
## 11     0    0    0    0      1    0    0    0      0  0    0      0    0
## 12     0    0    0    0      0    0    0    0      0  0    0      0    0
## 13     0    0    0    1      0    0    0    0      0  0    1      0    0
## 14     0    0    0    0      0    0    0    0      0  1    0      0    0
## 15     1    0    0    0      0    0    0    0      0  0    0      0    0
## 16     0    0    0    0      0    0    0    0      0  0    0      0    0
## 17     0    0    0    0      0    0    0    0      0  0    0      0    0
## 18     0    0    0    0      0    0    1    0      0  0    0      1    0
## 19     0    0    0    0      0    0    0    0      0  0    0      0    0
## 20     0    0    0    0      0    0    0    0      0  0    0      0    0
## 21     0    0    0    0      0    0    0    0      0  0    0      0    0
## 22     0    0    0    0      0    0    0    1      0  0    0      0    1
## 23     0    0    0    0      0    0    0    0      0  0    0      0    0
## 24     0    0    0    0      0    0    0    0      0  0    0      0    0
## 25     0    0    0    0      0    1    0    0      0  0    0      0    0
## 26     0    0    0    0      0    0    0    0      0  0    0      0    0
## 27     0    0    0    0      0    0    0    0      0  0    0      1    0
## 28     0    1    0    0      0    0    0    0      0  0    0      0    0
```

## 29	0	0	0	0	0	0	0	0	0	0	0	0	0
## 30	0	0	0	0	0	0	0	0	0	1	0	0	0
## 31	0	0	0	0	0	0	0	0	0	0	0	0	0
## 32	0	0	0	0	0	0	0	0	0	0	0	0	0
## 33	0	0	0	0	0	0	0	0	0	0	0	0	0
## 34	0	0	0	0	0	0	0	0	1	0	0	0	0
## 35	0	1	0	0	0	0	0	0	0	0	0	0	0
## 36	0	0	0	0	0	0	0	0	0	0	0	0	0
## 37	0	0	0	0	0	0	0	0	0	0	0	0	1
## 38	0	0	0	0	0	0	0	0	0	0	0	0	0
## 39	0	0	0	0	0	1	0	0	0	0	0	0	0
## 40	0	0	0	0	0	0	0	0	0	0	1	0	0
##	df\$Mood												
## 1	Angry												
## 2	Happy												
## 3	Angry												
## 4	Sad												
## 5	Happy												
## 6	Sad												
## 7	Normal												
## 8	Sad												
## 9	Happy												
## 10	Normal												
## 11	Normal												
## 12	Angry												
## 13	Angry												
## 14	Normal												
## 15	Happy												
## 16	Happy												
## 17	Angry												
## 18	Sad												
## 19	Normal												
## 20	Sad												
## 21	Happy												
## 22	Sad												
## 23	Angry												
## 24	Happy												
## 25	Normal												
## 26	Happy												
## 27	Sad												
## 28	Angry												
## 29	Normal												
## 30	Sad												
## 31	Happy												
## 32	Angry												
## 33	Normal												
## 34	Happy												
## 35	Sad												
## 36	Normal												
## 37	Sad												
## 38	Happy												
## 39	Angry												
## 40	Angry												

```

tdm_sparse <- removeSparseTerms(tdm,0.99)
tdm_dm2 <- as.data.frame(as.matrix(tdm_sparse)) # count matrix

tdm_df2 <- as.matrix((tdm_dm2 > 0) + 0) # binary instance matrix

tdm_df2 <- as.data.frame(tdm_df2)

tdm_df2 <- cbind(tdm_df2,df$Mood)

head(tdm_df2)

##      circumstances fast preapare rather full got hurrah marks made mistake chance
## 1           1      1      1      1      0      0      0      0      0      0
## 2           0      0      0      0      1      1      1      1      0      0
## 3           0      0      0      0      0      0      0      0      1      1
## 4           0      0      0      0      0      0      0      0      0      1
## 5           0      0      0      0      0      0      0      0      0      0
## 6           0      0      0      0      0      0      0      0      0      0
##      lost pass happy interview successful already dont leave please time will dad
## 1      0      0      0      0      0      0      0      0      0      0      0
## 2      0      0      0      0      0      0      0      0      0      0      0
## 3      0      0      0      0      0      0      0      0      0      0      0
## 4      1      1      0      0      0      0      0      0      0      0      0
## 5      0      0      1      1      1      0      0      0      0      0      0
## 6      1      0      0      0      0      1      1      1      1      0      0
##      didnt exam math sorry feeling good now pretty right coming tonight coffee
## 1      0      0      0      0      0      0      0      0      0      0      0
## 2      0      0      0      0      0      0      0      0      0      0      0
## 3      0      0      0      0      0      0      0      0      0      0      0
## 4      0      0      0      0      0      0      0      0      0      0      0
## 5      0      0      0      0      0      0      0      0      0      0      0
## 6      0      0      0      0      0      0      0      0      0      0      0
##      like morning till wait quiet talk can finallyi gift likes asked bit head one
## 1      0      0      0      0      0      0      0      0      0      0      0
## 2      0      0      0      0      0      0      0      0      0      0      0
## 3      0      0      0      0      0      0      0      0      0      0      0
## 4      0      0      0      0      0      0      0      0      0      0      0
## 5      0      0      0      0      0      0      0      0      0      0      0
## 6      0      0      0      0      0      0      0      0      0      0      0
##      question simple depressed english failed week call day career ill ruin feel
## 1      0      0      0      0      0      0      0      0      0      0      0
## 2      0      0      0      0      0      0      0      0      0      0      0
## 3      0      0      0      0      0      0      0      0      0      0      0
## 4      0      0      0      0      0      0      0      0      0      0      0
## 5      0      0      0      0      0      0      0      0      0      0      0
## 6      0      0      0      0      0      0      0      0      0      0      0
##      top wonderful world hear anger room went dream heaven job landed seventh fine
## 1      0      0      0      0      0      0      0      0      0      0      0
## 2      0      0      0      0      0      0      0      0      0      0      0
## 3      0      0      0      0      0      0      0      0      0      0      0
## 4      0      0      0      0      0      0      0      0      0      0      0
## 5      0      0      0      0      0      0      0      0      0      0      0
## 6      0      0      0      0      0      0      0      0      0      0      0
##      something smiling mind resting dead father month event excited man shouted

```

```
## 1      0      0      0      0      0      0      0      0      0      0      0
## 2      0      0      0      0      0      0      0      0      0      0      0
## 3      0      0      0      0      0      0      0      0      0      0      0
## 4      0      0      0      0      0      0      0      0      0      0      0
## 5      0      0      0      0      0      0      0      0      0      0      0
## 6      0      0      0      0      0      0      0      0      0      0      0
##      staff earth goes round relieved session crying puppy rakhi since chennai
## 1      0      0      0      0      0      0      0      0      0      0      0
## 2      0      0      0      0      0      0      0      0      0      0      0
## 3      0      0      0      0      0      0      0      0      0      0      0
## 4      0      0      0      0      0      0      0      0      0      0      0
## 5      0      0      0      0      0      0      0      0      0      0      0
## 6      0      0      0      0      0      0      0      0      0      0      0
##      travelling devastates loss know promotion rahul thrilled actions antisocial
## 1      0      0      0      0      0      0      0      0      0      0      0
## 2      0      0      0      0      0      0      0      0      0      0      0
## 3      0      0      0      0      0      0      0      0      0      0      0
## 4      0      0      0      0      0      0      0      0      0      0      0
## 5      0      0      0      0      0      0      0      0      0      0      0
## 6      0      0      0      0      0      0      0      0      0      0      0
##      sue dare elder way df$Mood
## 1      0      0      0      0      Angry
## 2      0      0      0      0      Happy
## 3      0      0      0      0      Angry
## 4      0      0      0      0      Sad
## 5      0      0      0      0      Happy
## 6      0      0      0      0      Sad
```

```
library(data.table)
setnames(tdm_df1,"df$Mood","Mood")
setnames(tdm_df2,"df$Mood","Mood")
```

```
tdm_df1$Mood<-as.factor(tdm_df1$Mood)
tdm_df2$Mood<-as.factor(tdm_df2$Mood)
```

```
table(tdm_df1$Mood)
```

```
##
##  Angry  Happy Normal   Sad
##      10     11      9    10
```

```
table(tdm_df1$Mood)
```

```
##
##  Angry  Happy Normal   Sad
##      10     11      9    10
```

Supervised Learning

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

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

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:NLP':
##
##      annotate
```

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

```
Naive Bayes Classifier
```

```
classifier_c2 <- naiveBayes(Mood ~ ., data = tdm_df2)
y_pred <- predict(classifier_c2, newdata = tdm_df2)
cm <- table(tdm_df2$Mood, y_pred)
confusionMatrix(cm)
```

```
## Confusion Matrix and Statistics
```

```
##
##           y_pred
##           Angry Happy Normal Sad
## Angry         0     0      10  0
## Happy         0     0      11  0
## Normal        0     0       9  0
## Sad           0     0      10  0
```

```
## Overall Statistics
```

```
##
##              Accuracy : 0.225
##              95% CI : (0.1084, 0.3845)
##      No Information Rate : 1
##      P-Value [Acc > NIR] : 1
##
##              Kappa : 0
```

```
## McNemar's Test P-Value : NA
```

```
## Statistics by Class:
```

```
##
##              Class: Angry Class: Happy Class: Normal Class: Sad
## Sensitivity              NA              NA              0.225      NA
## Specificity              0.75              0.725              NA      0.75
## Pos Pred Value              NA              NA              NA      NA
## Neg Pred Value              NA              NA              NA      NA
## Prevalence                0.00              0.000              1.000      0.00
## Detection Rate              0.00              0.000              0.225      0.00
## Detection Prevalence        0.25              0.275              0.225      0.25
## Balanced Accuracy              NA              NA              NA      NA
```

```
classifier_c1 <- naiveBayes(Mood ~ ., data = tdm_df1)
#classifier_c1
y_pred <- predict(classifier_c1, newdata = tdm_df1)
cm <- table(tdm_df1$Mood, y_pred)
confusionMatrix(cm)
```

```
## Confusion Matrix and Statistics
```

```
##
##           y_pred
##           Angry Happy Normal Sad
## Angry         3     6       1  0
## Happy         0    11       0  0
```



```
## Normal      0      5      4      0
## Sad         3      6      1      0
##
## Overall Statistics
##
##           Accuracy : 0.45
##           95% CI : (0.2926, 0.6151)
##           No Information Rate : 0.7
##           P-Value [Acc > NIR] : 0.999725
##
##           Kappa : 0.253
##
## McNemar's Test P-Value : 0.001211
##
## Statistics by Class:
##
##           Class: Angry Class: Happy Class: Normal Class: Sad
## Sensitivity           0.5000           0.3929           0.6667           NA
## Specificity           0.7941           1.0000           0.8529           0.75
## Pos Pred Value        0.3000           1.0000           0.4444           NA
## Neg Pred Value        0.9000           0.4138           0.9355           NA
## Prevalence            0.1500           0.7000           0.1500           0.00
## Detection Rate        0.0750           0.2750           0.1000           0.00
## Detection Prevalence  0.2500           0.2750           0.2250           0.25
## Balanced Accuracy      0.6471           0.6964           0.7598           NA
encode_ordinal <- function(x, order = unique(x)) {
  x <- as.numeric(factor(x, levels = order, exclude = NULL))
  x
}
```

Random Forest

```
library(randomForest)

## randomForest 4.7-1
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:ggplot2':
##
##      margin
rf <- randomForest(Mood ~ ., data = tdm_df1)
library(caret)
p1 <- predict(rf, tdm_df1)
```

```
p1

##      1      2      3      4      5      6      7      8      9     10     11
## Happy Happy Happy Sad Happy Sad Normal Sad Happy Happy Normal
##     12     13     14     15     16     17     18     19     20     21     22
## Happy Angry Normal Happy Happy Happy Sad Happy Happy Happy Sad
##     23     24     25     26     27     28     29     30     31     32     33
## Happy Happy Normal Happy Sad Sad Happy Normal Happy Happy Happy
```

```
##      34      35      36      37      38      39      40
## Happy   Sad Happy   Sad Happy Normal  Angry
## Levels: Angry Happy Normal Sad
```

```
confusionMatrix(p1,tdm_df1$Mood)
```

```
## Confusion Matrix and Statistics
```

```
##
```

```
##           Reference
```

```
## Prediction Angry Happy Normal Sad
```

```
##      Angry      2      0      0      0
```

```
##      Happy      6     11      5      1
```

```
##      Normal     1      0      4      1
```

```
##      Sad        1      0      0      8
```

```
##
```

```
## Overall Statistics
```

```
##
```

```
##           Accuracy : 0.625
```

```
##           95% CI : (0.458, 0.7727)
```

```
##      No Information Rate : 0.275
```

```
##      P-Value [Acc > NIR] : 3.953e-06
```

```
##
```

```
##           Kappa : 0.4928
```

```
##
```

```
##      McNemar's Test P-Value : 0.02026
```

```
##
```

```
## Statistics by Class:
```

```
##
```

```
##           Class: Angry Class: Happy Class: Normal Class: Sad
```

```
## Sensitivity      0.2000      1.0000      0.4444      0.8000
```

```
## Specificity      1.0000      0.5862      0.9355      0.9667
```

```
## Pos Pred Value    1.0000      0.4783      0.6667      0.8889
```

```
## Neg Pred Value    0.7895      1.0000      0.8529      0.9355
```

```
## Prevalence        0.2500      0.2750      0.2250      0.2500
```

```
## Detection Rate    0.0500      0.2750      0.1000      0.2000
```

```
## Detection Prevalence 0.0500      0.5750      0.1500      0.2250
```

```
## Balanced Accuracy  0.6000      0.7931      0.6900      0.8833
```

```
library(randomForest)
```

```
rf2 <- randomForest(Mood ~ ., data = tdm_df2)
```

```
library(caret)
```

```
p2<-predict(rf2,tdm_df2)
```

```
confusionMatrix(p2,tdm_df2$Mood)
```

```
## Confusion Matrix and Statistics
```

```
##
```

```
##           Reference
```

```
## Prediction Angry Happy Normal Sad
```

```
##      Angry      10      0      0      0
```

```
##      Happy      0     11      0      0
```

```
##      Normal     0      0      9      0
```

```
##      Sad        0      0      0     10
```

```
##
```

```
## Overall Statistics
```

```
##
```

```
## Accuracy : 1
## 95% CI : (0.9119, 1)
## No Information Rate : 0.275
## P-Value [Acc > NIR] : < 2.2e-16
##
## Kappa : 1
##
## McNemar's Test P-Value : NA
##
## Statistics by Class:
##
## Class: Angry Class: Happy Class: Normal Class: Sad
## Sensitivity 1.00 1.000 1.000 1.00
## Specificity 1.00 1.000 1.000 1.00
## Pos Pred Value 1.00 1.000 1.000 1.00
## Neg Pred Value 1.00 1.000 1.000 1.00
## Prevalence 0.25 0.275 0.225 0.25
## Detection Rate 0.25 0.275 0.225 0.25
## Detection Prevalence 0.25 0.275 0.225 0.25
## Balanced Accuracy 1.00 1.000 1.000 1.00
```

```
library(e1071)
library(caTools)
library(class)
```

KNN Classifier

```
kdf1<-tdm_df1
kdf2<-tdm_df2
kdf1$Mood<-encode_ordinal(kdf1$Mood)
kdf2$Mood<-encode_ordinal(kdf2$Mood)
```

```
str(kdf1)
```

```
## 'data.frame': 40 obs. of 14 variables:
## $ got : num 0 1 0 0 0 0 0 0 0 0 ...
## $ lost : num 0 0 0 1 0 1 0 0 0 0 ...
## $ pass : num 0 0 0 1 0 0 0 1 0 0 ...
## $ dont : num 0 0 0 0 0 1 0 0 0 0 ...
## $ please : num 0 0 0 0 0 1 0 0 0 0 ...
## $ will : num 0 0 0 0 0 0 1 0 0 0 ...
## $ exam : num 0 0 0 0 0 0 0 1 0 0 ...
## $ sorry : num 0 0 0 0 0 0 0 1 0 0 ...
## $ feeling: num 0 0 0 0 0 0 0 0 1 0 ...
## $ now : num 0 0 0 0 0 0 0 0 1 0 ...
## $ talk : num 0 0 0 0 0 0 0 0 0 0 ...
## $ failed : num 0 0 0 0 0 0 0 0 0 0 ...
## $ hear : num 0 0 0 0 0 0 0 0 0 0 ...
## $ Mood : num 1 2 1 3 2 3 4 3 2 4 ...
```

```
classifier_knn1 <- knn(train = kdf1,
  test = kdf1,
  cl = kdf1$Mood,
  k = 4)
# Confusion Matrix
cm <- table(kdf1$Mood, classifier_knn1)
```

```

cm

##      classifier_knn1
##      1  2  3  4
##    1 10  0  0  0
##    2  0 11  0  0
##    3  0  7  3  0
##    4  0  0  0  9

#confusionMatrix(kdf$Mood,classifier_knn)
# Calculate out of Sample error
misClassError1 <- mean(classifier_knn1 != kdf1$Mood)
print(paste('Accuracy =', 1-misClassError1))

## [1] "Accuracy = 0.825"

classifier_knn2 <- knn(train = kdf2,
                      test = kdf2,
                      cl = kdf2$Mood,
                      k = 4)

# Confusiin Matrix
cm2 <- table(kdf2$Mood, classifier_knn2)
cm2

##      classifier_knn2
##      1  2  3  4
##    1 10  0  0  0
##    2  0 11  0  0
##    3  0  0 10  0
##    4  0  0  0  9

misClassError2 <- mean(classifier_knn2 != kdf2$Mood)
print(paste('Accuracy =', 1-misClassError2))

## [1] "Accuracy = 1"

Decision Tree Classifier

library(party)

## Loading required package: grid
## Loading required package: mvtnorm
## Loading required package: modeltools
## Loading required package: stats4
## Loading required package: strucchange
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##      as.Date, as.Date.numeric
## Loading required package: sandwich

```

```
Tree1<- ctree(Mood ~ .,data = tdm_df1)
#plot(Tree)
```

```
tree_pred1<-predict(Tree1,tdm_df1)
m_at1<- table(tdm_df1$Mood,tree_pred1)
m_at1
```

```
##           tree_pred1
##           Angry Happy Normal Sad
##   Angry           0    10      0  0
##   Happy           0    11      0  0
##   Normal          0     9      0  0
##   Sad             0    10      0  0
```

```
ac_test1<-sum(diag(m_at1))/sum(m_at1)
print(paste('Accuracy:',ac_test1))
```

```
## [1] "Accuracy: 0.275"
```

```
Tree2<- ctree(Mood ~ .,data = tdm_df2)
tree_pred2<-predict(Tree2,tdm_df2)
m_at2<- table(tdm_df2$Mood,tree_pred2)
m_at2
```

```
##           tree_pred2
##           Angry Happy Normal Sad
##   Angry           0    10      0  0
##   Happy           0    11      0  0
##   Normal          0     9      0  0
##   Sad             0    10      0  0
```

```
ac_test2<-sum(diag(m_at2))/sum(m_at2)
print(paste('Accuracy:',ac_test2))
```

```
## [1] "Accuracy: 0.275"
```

XG Boost Classifier

```
library(xgboost)
```

```
X_train = data.matrix(tdm_df2[, -110])
y_train = tdm_df2[, 110]
X_train2= data.matrix(tdm_df1[, -14])
y_train2=tdm_df1[, 14]
```

```
xgboost_train = xgb.DMatrix(data=X_train, label=y_train)
```

```
modelXG <- xgboost(data = xgboost_train,           # the data
                    max.depth=4,                    # max depth
                    nrounds=50)
```

```
## [1] train-rmse:1.748561
## [2] train-rmse:1.392624
## [3] train-rmse:1.160464
## [4] train-rmse:1.003893
## [5] train-rmse:0.888453
## [6] train-rmse:0.788660
## [7] train-rmse:0.724026
```

```
## [8] train-rmse:0.673720
## [9] train-rmse:0.634105
## [10] train-rmse:0.599409
## [11] train-rmse:0.571362
## [12] train-rmse:0.546756
## [13] train-rmse:0.527502
## [14] train-rmse:0.503119
## [15] train-rmse:0.486725
## [16] train-rmse:0.470177
## [17] train-rmse:0.455816
## [18] train-rmse:0.441508
## [19] train-rmse:0.427086
## [20] train-rmse:0.410613
## [21] train-rmse:0.399094
## [22] train-rmse:0.387176
## [23] train-rmse:0.375039
## [24] train-rmse:0.364386
## [25] train-rmse:0.350964
## [26] train-rmse:0.340824
## [27] train-rmse:0.330182
## [28] train-rmse:0.320526
## [29] train-rmse:0.309195
## [30] train-rmse:0.299928
## [31] train-rmse:0.290739
## [32] train-rmse:0.282673
## [33] train-rmse:0.274694
## [34] train-rmse:0.266113
## [35] train-rmse:0.258720
## [36] train-rmse:0.251812
## [37] train-rmse:0.244903
## [38] train-rmse:0.237934
## [39] train-rmse:0.231043
## [40] train-rmse:0.224886
## [41] train-rmse:0.218043
## [42] train-rmse:0.211828
## [43] train-rmse:0.205009
## [44] train-rmse:0.199479
## [45] train-rmse:0.194155
## [46] train-rmse:0.188439
## [47] train-rmse:0.183208
## [48] train-rmse:0.177938
## [49] train-rmse:0.173145
## [50] train-rmse:0.168390
```

```
summary(modelXG)
```

##	Length	Class	Mode
## handle	1	xgb.Booster.handle	externalptr
## raw	52940	-none-	raw
## niter	1	-none-	numeric
## evaluation_log	2	data.table	list
## call	14	-none-	call
## params	2	-none-	list
## callbacks	2	-none-	list
## feature_names	109	-none-	character

```

## nfeatures          1 -none-          numeric
pred_test = predict(modelXG, xgboost_train)
pred_test

## [1] 1.171546 2.049056 1.171546 3.944915 2.165642 3.873734 2.767050 3.778819
## [9] 2.151987 2.789443 2.885432 1.163879 1.037458 2.785636 2.049056 2.165642
## [17] 1.171984 4.000412 2.790685 3.777169 2.165642 3.956551 1.171984 2.165642
## [25] 2.805194 2.165642 3.765035 1.165349 2.793993 3.769371 2.165642 1.177595
## [33] 2.794769 2.049056 3.786432 2.798812 3.801840 2.165642 1.167576 1.037458

pred_test[(pred_test>4)] = 4
pred_y = as.factor((levels(y_train))[round(pred_test)])
print(pred_y)

## [1] Angry Happy Angry Sad Happy Sad Normal Sad Happy Normal
## [11] Normal Angry Angry Normal Happy Happy Angry Sad Normal Sad
## [21] Happy Sad Angry Happy Normal Happy Sad Angry Normal Sad
## [31] Happy Angry Normal Happy Sad Normal Sad Happy Angry Angry
## Levels: Angry Happy Normal Sad

conf_mat = confusionMatrix(y_train, pred_y)
print(conf_mat)

## Confusion Matrix and Statistics
##
##              Reference
## Prediction Angry Happy Normal Sad
##      Angry      10      0      0      0
##      Happy       0     11      0      0
##      Normal       0      0      9      0
##      Sad          0      0      0     10
##
## Overall Statistics
##
##              Accuracy : 1
##              95% CI : (0.9119, 1)
##      No Information Rate : 0.275
##      P-Value [Acc > NIR] : < 2.2e-16
##
##              Kappa : 1
##
##      McNemar's Test P-Value : NA
##
## Statistics by Class:
##
##              Class: Angry Class: Happy Class: Normal Class: Sad
## Sensitivity              1.00          1.000          1.000          1.00
## Specificity              1.00          1.000          1.000          1.00
## Pos Pred Value           1.00          1.000          1.000          1.00
## Neg Pred Value           1.00          1.000          1.000          1.00
## Prevalence               0.25          0.275          0.225          0.25
## Detection Rate           0.25          0.275          0.225          0.25
## Detection Prevalence     0.25          0.275          0.225          0.25
## Balanced Accuracy        1.00          1.000          1.000          1.00

```

```

xgboost_train2 = xgb.DMatrix(data=X_train2, label=y_train2)
modelXG2 <- xgboost(data = xgboost_train2,           # the data
                    max.depth=4,                     # max depth
                    nrounds=50)

```

```

## [1] train-rmse:1.774120
## [2] train-rmse:1.463091
## [3] train-rmse:1.255274
## [4] train-rmse:1.109312
## [5] train-rmse:1.007167
## [6] train-rmse:0.937215
## [7] train-rmse:0.884328
## [8] train-rmse:0.847492
## [9] train-rmse:0.822527
## [10] train-rmse:0.805277
## [11] train-rmse:0.792765
## [12] train-rmse:0.783828
## [13] train-rmse:0.776290
## [14] train-rmse:0.771643
## [15] train-rmse:0.768203
## [16] train-rmse:0.765704
## [17] train-rmse:0.763781
## [18] train-rmse:0.761378
## [19] train-rmse:0.760154
## [20] train-rmse:0.759363
## [21] train-rmse:0.758574
## [22] train-rmse:0.757909
## [23] train-rmse:0.757369
## [24] train-rmse:0.756265
## [25] train-rmse:0.755860
## [26] train-rmse:0.755535
## [27] train-rmse:0.755261
## [28] train-rmse:0.755021
## [29] train-rmse:0.754838
## [30] train-rmse:0.754671
## [31] train-rmse:0.754008
## [32] train-rmse:0.753885
## [33] train-rmse:0.753767
## [34] train-rmse:0.753675
## [35] train-rmse:0.753602
## [36] train-rmse:0.753175
## [37] train-rmse:0.753116
## [38] train-rmse:0.753073
## [39] train-rmse:0.752774
## [40] train-rmse:0.752732
## [41] train-rmse:0.752708
## [42] train-rmse:0.752679
## [43] train-rmse:0.752659
## [44] train-rmse:0.752642
## [45] train-rmse:0.752624
## [46] train-rmse:0.752613
## [47] train-rmse:0.752420
## [48] train-rmse:0.752410
## [49] train-rmse:0.752399

```



```
## [50] train-rmse:0.752391
```

```
pred_test2 = predict(modelXG2, xgboost_train2)
pred_test2[(pred_test2>4)] = 4
pred_y2 = as.factor((levels(y_train2))[round(pred_test2)])
conf_mat2 = confusionMatrix(y_train2, pred_y2)
print(conf_mat2)
```

```
## Confusion Matrix and Statistics
```

```
##
```

```
##           Reference
```

```
## Prediction Angry Happy Normal Sad
```

```
##    Angry      2      8      0      0
```

```
##    Happy      0     11      0      0
```

```
##   Normal      0      7      2      0
```

```
##     Sad       0      2      1      7
```

```
##
```

```
## Overall Statistics
```

```
##
```

```
##           Accuracy : 0.55
```

```
##           95% CI : (0.3849, 0.7074)
```

```
##    No Information Rate : 0.7
```

```
##    P-Value [Acc > NIR] : 0.9852
```

```
##
```

```
##           Kappa : 0.3872
```

```
##
```

```
## McNemar's Test P-Value : NA
```

```
##
```

```
## Statistics by Class:
```

```
##
```

```
##           Class: Angry Class: Happy Class: Normal Class: Sad
```

```
## Sensitivity           1.0000           0.3929           0.6667           1.0000
```

```
## Specificity           0.7895           1.0000           0.8108           0.9091
```

```
## Pos Pred Value        0.2000           1.0000           0.2222           0.7000
```

```
## Neg Pred Value        1.0000           0.4138           0.9677           1.0000
```

```
## Prevalence            0.0500           0.7000           0.0750           0.1750
```

```
## Detection Rate        0.0500           0.2750           0.0500           0.1750
```

```
## Detection Prevalence  0.2500           0.2750           0.2250           0.2500
```

```
## Balanced Accuracy     0.8947           0.6964           0.7387           0.9545
```

```
Unsupervised Learning
```

```
Kmeans Clustering
```

```
kdata2<-tdm_df2[, -110]
```

```
kdata1<-tdm_df1[, -14]
```

```
head(kdata2)
```

```
##      circumstances fast preapare rather full got hurrah marks made mistake chance
```

```
## 1           1      1      1      1      0      0      0      0      0      0      0
```

```
## 2           0      0      0      0      1      1      1      1      0      0      0
```

```
## 3           0      0      0      0      0      0      0      0      1      1      0
```

```
## 4           0      0      0      0      0      0      0      0      0      0      1
```

```
## 5           0      0      0      0      0      0      0      0      0      0      0
```

```
## 6           0      0      0      0      0      0      0      0      0      0      0
```

```
##      lost pass happy interview successful already dont leave please time will dad
```

## 1	0	0	0	0	0	0	0	0	0	0	0	0		
## 2	0	0	0	0	0	0	0	0	0	0	0	0		
## 3	0	0	0	0	0	0	0	0	0	0	0	0		
## 4	1	1	0	0	0	0	0	0	0	0	0	0		
## 5	0	0	1	1	1	0	0	0	0	0	0	0		
## 6	1	0	0	0	0	1	1	1	1	0	0	0		
##	didnt	exam	math	sorry	feeling	good	now	pretty	right	coming	tonight	coffee		
## 1	0	0	0	0	0	0	0	0	0	0	0	0		
## 2	0	0	0	0	0	0	0	0	0	0	0	0		
## 3	0	0	0	0	0	0	0	0	0	0	0	0		
## 4	0	0	0	0	0	0	0	0	0	0	0	0		
## 5	0	0	0	0	0	0	0	0	0	0	0	0		
## 6	0	0	0	0	0	0	0	0	0	0	0	0		
##	like	morning	till	wait	quiet	talk	can	finallyi	gift	likes	asked	bit	head	one
## 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	question	simple	depressed	english	failed	week	call	day	career	ill	ruin	feel		
## 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	top	wonderful	world	hear	anger	room	went	dream	heaven	job	landed	seventh	fine	
## 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	something	smiling	mind	resting	dead	father	month	event	excited	man	shouted			
## 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	staff	earth	goes	round	relieved	session	crying	puppy	rakhi	since	chennai			
## 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	travelling	devastates	loss	know	promotion	rahul	thrilled	actions	antisocial					
## 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0

```
## 6      0      0      0      0      0      0      0      0      0
##  sue dare elder way
## 1      0      0      0      0
## 2      0      0      0      0
## 3      0      0      0      0
## 4      0      0      0      0
## 5      0      0      0      0
## 6      0      0      0      0
```

```
kmean2<-kmeans(kdata2,4)
kmean2
```

```
## K-means clustering with 4 clusters of sizes 31, 3, 4, 2
```

```
##
```

```
## Cluster means:
```

```
##      circumstances      fast      preapare      rather      full      got
## 1      0.03225806 0.03225806 0.03225806 0.03225806 0.03225806 0.06451613
## 2      0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## 3      0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## 4      0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##      hurrah      marks      made      mistake chance lost pass      happy
## 1 0.03225806 0.03225806 0.03225806 0.03225806 0.00 0 0.00 0.03225806
## 2 0.00000000 0.00000000 0.00000000 0.00000000 0.00 0 0.00 0.00000000
## 3 0.00000000 0.00000000 0.00000000 0.00000000 0.25 1 0.25 0.00000000
## 4 0.00000000 0.00000000 0.00000000 0.00000000 0.00 0 0.50 0.00000000
##      interview successful already      dont leave      please      time
## 1 0.03225806 0.03225806 0.00 0.03225806 0.00 0.03225806 0.03225806
## 2 0.00000000 0.00000000 0.00 0.00000000 0.00 0.00000000 0.00000000
## 3 0.00000000 0.00000000 0.25 0.25000000 0.25 0.25000000 0.00000000
## 4 0.00000000 0.00000000 0.00 0.00000000 0.00 0.00000000 0.00000000
##      will dad didnt exam math      sorry      feeling      good now      pretty
## 1 0.09677419 0.0 0.0 0 0.0 0.03225806 0.03225806 0.00000000 0 0.00000000
## 2 0.00000000 0.0 0.0 0 0.0 0.00000000 0.33333333 0.33333333 1 0.33333333
## 3 0.00000000 0.0 0.0 0 0.0 0.00000000 0.00000000 0.00000000 0 0.00000000
## 4 0.00000000 0.5 0.5 1 0.5 0.50000000 0.00000000 0.00000000 0 0.00000000
##      right      coming      tonight      coffee      like      morning      till
## 1 0.00000000 0.03225806 0.03225806 0.03225806 0.03225806 0.03225806 0.03225806
## 2 0.33333333 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## 3 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## 4 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##      wait      quiet      talk      can      finallyi      gift      likes
## 1 0.03225806 0.03225806 0.06451613 0.00000000 0.03225806 0.03225806 0.03225806
## 2 0.00000000 0.00000000 0.00000000 0.33333333 0.00000000 0.00000000 0.00000000
## 3 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## 4 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##      asked      bit      head      one      question      simple depressed
## 1 0.03225806 0.03225806 0.03225806 0.03225806 0.03225806 0.03225806 0.0
## 2 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.0
## 3 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.0
## 4 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.5
##      english      failed week      call      day      career      ill
## 1      0.0 0.03225806 0.0 0.03225806 0.03225806 0.03225806 0.03225806
## 2      0.0 0.00000000 0.0 0.00000000 0.00000000 0.00000000 0.00000000
## 3      0.0 0.00000000 0.0 0.00000000 0.00000000 0.00000000 0.00000000
## 4      0.5 0.50000000 0.5 0.00000000 0.00000000 0.00000000 0.00000000
```

```

##      ruin      feel      top wonderful      world      hear      anger
## 1 0.03225806 0.03225806 0.03225806 0.03225806 0.03225806 0.06451613 0.03225806
## 2 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## 3 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## 4 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##      room      went      dream      heaven      job      landed      seventh
## 1 0.03225806 0.03225806 0.03225806 0.03225806 0.03225806 0.03225806 0.03225806
## 2 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## 3 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## 4 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##      fine something      smiling mind      resting      dead      father
## 1 0.03225806 0.03225806 0.03225806 0.00 0.03225806 0.00000000 0.00000000
## 2 0.00000000 0.00000000 0.00000000 0.00 0.00000000 0.3333333 0.3333333
## 3 0.00000000 0.00000000 0.00000000 0.25 0.00000000 0.00000000 0.00000000
## 4 0.00000000 0.00000000 0.00000000 0.00 0.00000000 0.00000000 0.00000000
##      month      event      excited      man      shouted      staff      earth
## 1 0.00000000 0.03225806 0.03225806 0.03225806 0.03225806 0.03225806 0.03225806
## 2 0.3333333 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## 3 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## 4 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##      goes      round      relieved      session crying puppy rakhi since
## 1 0.03225806 0.03225806 0.03225806 0.03225806 0.00 0.00 0.00 0.00
## 2 0.00000000 0.00000000 0.00000000 0.00000000 0.00 0.00 0.00 0.00
## 3 0.00000000 0.00000000 0.00000000 0.00000000 0.25 0.25 0.25 0.25
## 4 0.00000000 0.00000000 0.00000000 0.00000000 0.00 0.00 0.00 0.00
##      chennai travelling devastates      loss      know      promotion      rahul
## 1 0.03225806 0.03225806 0.03225806 0.03225806 0.03225806 0.03225806 0.03225806
## 2 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## 3 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## 4 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##      thrilled      actions antisocial      sue      dare      elder      way
## 1 0.03225806 0.03225806 0.03225806 0.03225806 0.03225806 0.03225806 0.03225806
## 2 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## 3 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## 4 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
##
## Clustering vector:
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
## 1 1 1 3 1 3 1 4 2 1 1 1 1 2 1 1 1 4 1 1 1 1 1 1 1
## 27 28 29 30 31 32 33 34 35 36 37 38 39 40
## 1 3 1 2 1 1 1 1 3 1 1 1 1 1
##
## Within cluster sum of squares by cluster:
## [1] 85.741935 5.333333 8.250000 4.500000
## (between_SS / total_SS = 14.7 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
##
## #kmean$cluster
m_at2<- table(tdm_df2$Mood,kmean2$cluster)
m_at2

```

```
##
##      1  2  3  4
##  Angry  9  0  1  0
##  Happy 10  1  0  0
##  Normal  8  1  0  0
##  Sad    4  1  3  2

ac_test2<-sum(diag(m_at2))/sum(m_at2)
print(paste('Accuracy:',ac_test2))

## [1] "Accuracy: 0.3"

kmean<-kmeans(kdata1,4)
kmean

## K-means clustering with 4 clusters of sizes 3, 2, 31, 4
##
## Cluster means:
##      got lost      pass dont      please      will      exam      sorry
## 1 0.00000000  0 0.3333333 0.00 0.00000000 0.00000000 0.6666667 0.6666667
## 2 0.00000000  0 0.0000000 0.50 0.00000000 0.00000000 0.0000000 0.0000000
## 3 0.06451613  0 0.0000000 0.00 0.03225806 0.09677419 0.0000000 0.0000000
## 4 0.00000000  1 0.2500000 0.25 0.25000000 0.00000000 0.0000000 0.0000000
##      feeling      now talk      failed      hear
## 1 0.00000000 0.00000000  0 0.33333333 0.33333333
## 2 0.00000000 0.00000000  1 0.00000000 0.00000000
## 3 0.06451613 0.09677419  0 0.03225806 0.03225806
## 4 0.00000000 0.00000000  0 0.00000000 0.00000000
##
## Clustering vector:
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
##  3  3  3  4  3  4  3  1  3  3  3  3  2  3  3  3  3  1  3  3  3  1  3  3  3  3
## 27 28 29 30 31 32 33 34 35 36 37 38 39 40
##  3  4  3  3  3  3  3  3  4  3  3  3  3  2
##
## Within cluster sum of squares by cluster:
## [1] 3.333333 0.500000 12.064516 2.250000
## (between_SS / total_SS = 35.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
m_at<- table(tdm_df1$Mood,kmean$cluster)
m_at

##
##      1  2  3  4
##  Angry  0  2  7  1
##  Happy  0  0 11  0
##  Normal  0  0  9  0
##  Sad    3  0  4  3

ac_test<-sum(diag(m_at))/sum(m_at)
print(paste('Accuracy:',ac_test))
```

```
## [1] "Accuracy: 0.3"
```

K-Medoid Clustering

```
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(class)
```

```
library(cluster)
```

```
kmedoid<-pam(kdata1,metric = "manhattan",4)
```

```
kmedoid<-kmedoid$clustering
```

```
cm_med<- table(tdm_df1$Mood,kmedoid)
```

```
cm_med
```

```
##           kmedoid
##           1  2  3  4
## Angry      8  1  1  0
## Happy     10  0  0  1
## Normal      6  0  2  1
## Sad         6  3  0  1
```

```
ac_test_med<-sum(diag(cm_med))/sum(cm_med)
```

```
print(paste('Accuracy:',ac_test_med))
```

```
## [1] "Accuracy: 0.275"
```

```
kmedoid2<-pam(kdata2,metric = "manhattan",4)
```

```
kmedoid2<-kmedoid2$clustering
```

```
cm_med2<- table(tdm_df2$Mood,kmedoid2)
```

```
cm_med2
```

```
##           kmedoid2
##           1  2  3  4
## Angry      9  0  1  0
## Happy     11  0  0  0
## Normal      9  0  0  0
## Sad         8  1  0  1
```

```
ac_test_med2<-sum(diag(cm_med2))/sum(cm_med2)
```

```
print(paste('Accuracy:',ac_test_med2))
```

```
## [1] "Accuracy: 0.25"
```

```
Type<-c("Supervised","Supervised","Supervised","Supervised","Supervised","Unsupervised","Unsupervised")
```

```
Algorithm<-c("Naive Bayes","Random Forest","KNN Classifier","Decision Tree","XG Boost","Kmeans","Kmedoid")
```

```
Accuracy1<-c(0.45,0.6,0.9,0.275,0.55,ac_test,ac_test_med)
```

```
Accuracy2<-c(0.225,1,1,0.275,1,ac_test2,ac_test_med2)
```

```
Models<-data.frame(Type,Algorithm,Accuracy1,Accuracy2)
```

```
Models
```

```
##           Type      Algorithm Accuracy1 Accuracy2
## 1  Supervised   Naive Bayes    0.450    0.225
## 2  Supervised Random Forest    0.600    1.000
## 3  Supervised KNN Classifier    0.900    1.000
## 4  Supervised Decision Tree    0.275    0.275
```

## 5	Supervised	XG Boost	0.550	1.000
## 6	Unsupervised	Kmeans	0.300	0.300
## 7	Unsupervised	Kmedoid	0.275	0.250