School of Information Technology & Engineering (SITE)

MTech Software Engineering

Soft computing(SWE1011)

Sentimental analysis of twitter data(apple phone review)

Name:- Bhargava k

RegNo:-15MIS0252

Slot:B2+TB2

Submited to:-

Chandrasegar T

1)Raw dataset

Link:-
https://www.kaggle.com/c/apple-computers-twitter-sentiment2/data

https://drive.google.com/file/d/0B5W8CO0Gb2GGTEs3SUZ0Qnp3Mms/view

No of attributes:-17

No of Inputs:-

Here we are mainly focusing the text input from the user becouse we are analysis through the text rest of them are neglated

1) Text

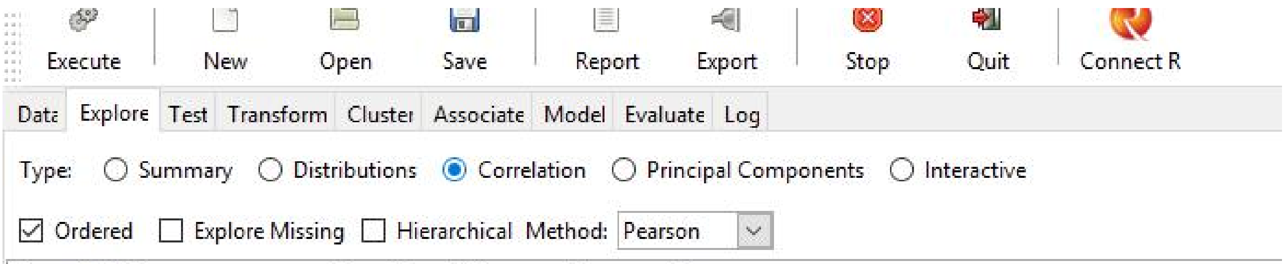
Algorithm used:-
For the twitter data analysis we can commannly used the sentimental analysis becouse how the users are positely, negatively..etc

1) Sentimental analysis

Abstract:-

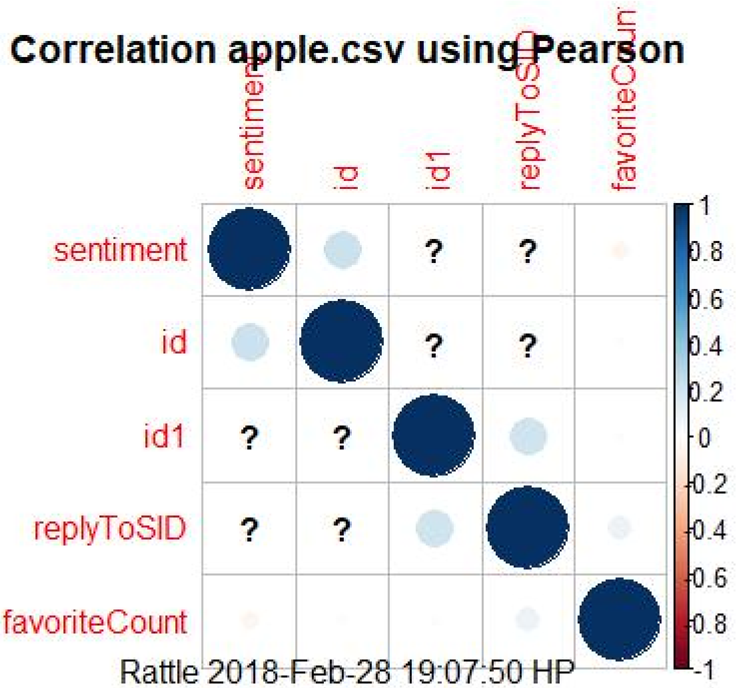
Twitter data analysis we can analysis all kinds of information like(phones, politics,movies..etc) these we can analysis various techniques how peoples are in the positely or nagitively here I am analysis the "apple phones" doing the project This is the look into the sentiment around the Apple phones on tweets like the text. Sentiment Analysis is the process of 'computationally' determining whether a piece of writing is positive, negative or neutral. It's also known as opinion mining, deriving the opinion or attitude of a speaker.it's largely due to the massive amount of data that can be collected from a single Tweet.

Dataextraction:-




```
Correlation summary using the 'Pearson' covariance.

Note that only correlations between numeric variables are reported.

                 sentiment         id         id1 replyToSID favoriteCount
sentiment       1.00000000  0.2167322          NA         NA   -0.05092203
id              0.21673217  1.0000000          NA         NA   -0.02170610
id1                     NA         NA 1.00000000 0.20597823    0.01369044
replyToSID              NA         NA 0.20597823 1.00000000    0.08695441
favoriteCount -0.05092203 -0.0217061 0.01369044 0.08695441    1.00000000

Rattle timestamp: 2018-02-28 19:07:50 HP
======================================================================
```

Correlation graph:-



3) models:-

1) tree model:-

Data Explore Test Transform Cluster Associate Model Evaluate Log

Type: (•) Tree ( ) Forest ( ) Boost ( ) SVM ( ) Linear ( ) Neural Net ( ) Survival ( ) All

Target: sentiment Algorithm: (•) Traditional ( ) Conditional

Min Split: 20 Max Depth: 3 Priors:

Min Bucket: 7 Complexity: 0.0100 Loss Matrix:

```
Summary of the Decision Tree model for Classification (built using 'rpart'):

n=83 (617 observations deleted due to missingness)

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 83 36 3 (0.32530120 0.56626506 0.10843373)
   2) id< 6.234994e+08 76 30 3 (0.31578947 0.60526316 0.07894737)
     4) id< 6.234994e+08 61 29 3 (0.39344262 0.52459016 0.08196721)
       8) id>=6.234993e+08 46 24 1 (0.47826087 0.47826087 0.04347826)
        16) id< 6.234993e+08 13  4 1 (0.69230769 0.30769231 0.00000000) *
        17) id>=6.234993e+08 33 15 3 (0.39393939 0.54545455 0.06060606)
          34) id>=6.234994e+08 10  3 1 (0.70000000 0.30000000 0.00000000) *
          35) id< 6.234994e+08 23  8 3 (0.26086957 0.65217391 0.08695652) *
       9) id< 6.234993e+08 15  5 3 (0.13333333 0.66666667 0.20000000) *
     5) id>=6.234994e+08 15  1 3 (0.00000000 0.93333333 0.06666667) *
   3) id>=6.234994e+08 7  4 1 (0.42857143 0.14285714 0.42857143) *

Classification tree:
rpart(formula = sentiment ~ ., data = crs$dataset[crs$train,
    c(crs$input, crs$target)], method = "class", parms = list(split = "information"),
    control = rpart.control(usesurrogate = 0, maxsurrogate = 0))

Variables actually used in tree construction:
[1] id
```

Data Explore Test Transform Cluster Associate Model Evaluate Log

Type: (•) Tree ( ) Forest ( ) Boost ( ) SVM ( ) Linear ( ) Neural Net ( ) Survival ( ) All

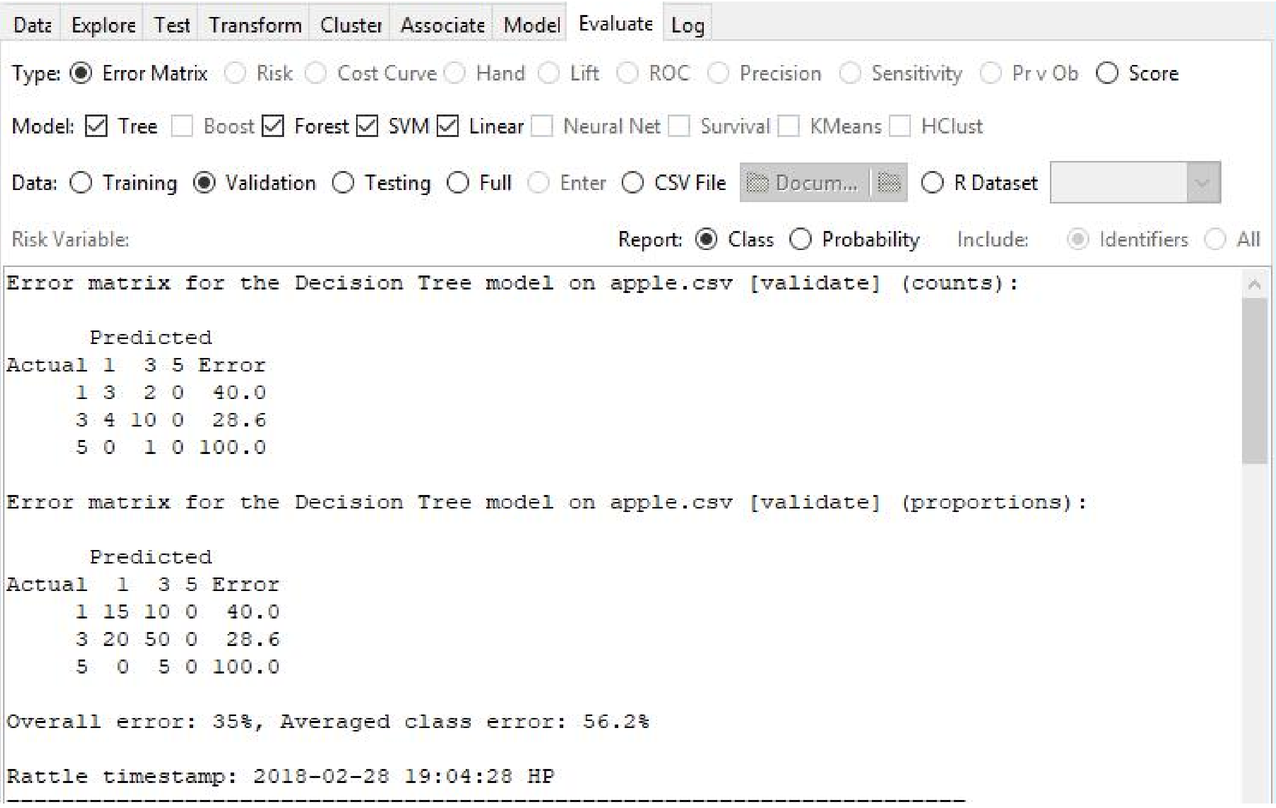
Target: sentiment Algorithm: (•) Traditional ( ) Conditional

Min Split: 20 Max Depth: 3 Priors:

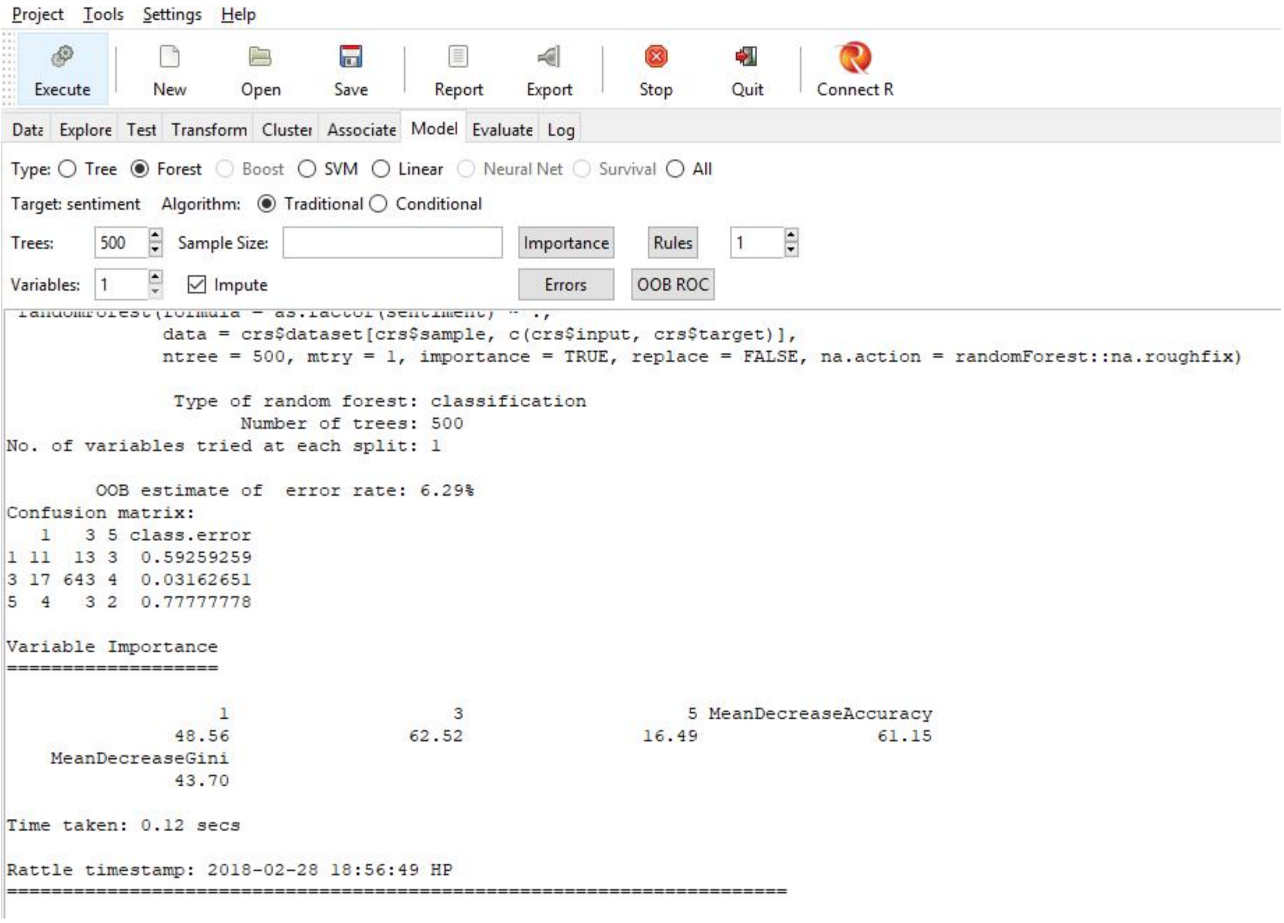
Min Bucket: 7 Complexity: 0.0100 Loss Matrix:

```
          35) id< 6.234994e+08 23  8 3 (0.26086957 0.65217391 0.08695652) *
       9) id< 6.234993e+08 15  5 3 (0.13333333 0.66666667 0.20000000) *
     5) id>=6.234994e+08 15  1 3 (0.00000000 0.93333333 0.06666667) *
   3) id>=6.234994e+08 7  4 1 (0.42857143 0.14285714 0.42857143) *

Classification tree:
rpart(formula = sentiment ~ ., data = crs$dataset[crs$train,
    c(crs$input, crs$target)], method = "class", parms = list(split = "information"),
    control = rpart.control(usesurrogate = 0, maxsurrogate = 0))

Variables actually used in tree construction:
[1] id

Root node error: 36/83 = 0.43373

n=83 (617 observations deleted due to missingness)

        CP nsplit rel error xerror    xstd
1 0.055556      0   1.00000 1.0000 0.12542
2 0.046296      1   0.94444 1.1389 0.12652
3 0.010000      5   0.69444 1.0000 0.12542

Time taken: 0.01 secs

Rattle timestamp: 2018-02-28 18:54:31 HP
```

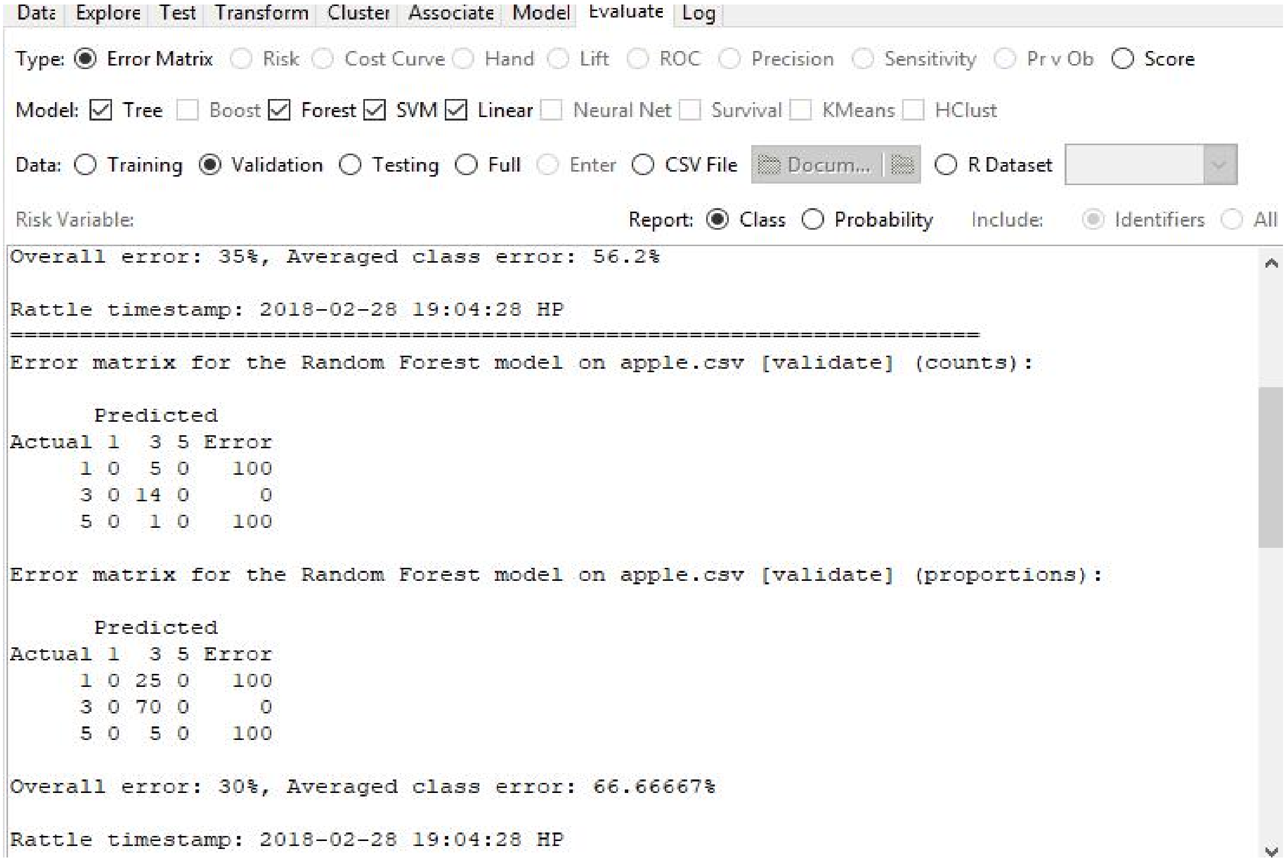
Error matrix for tree model:-



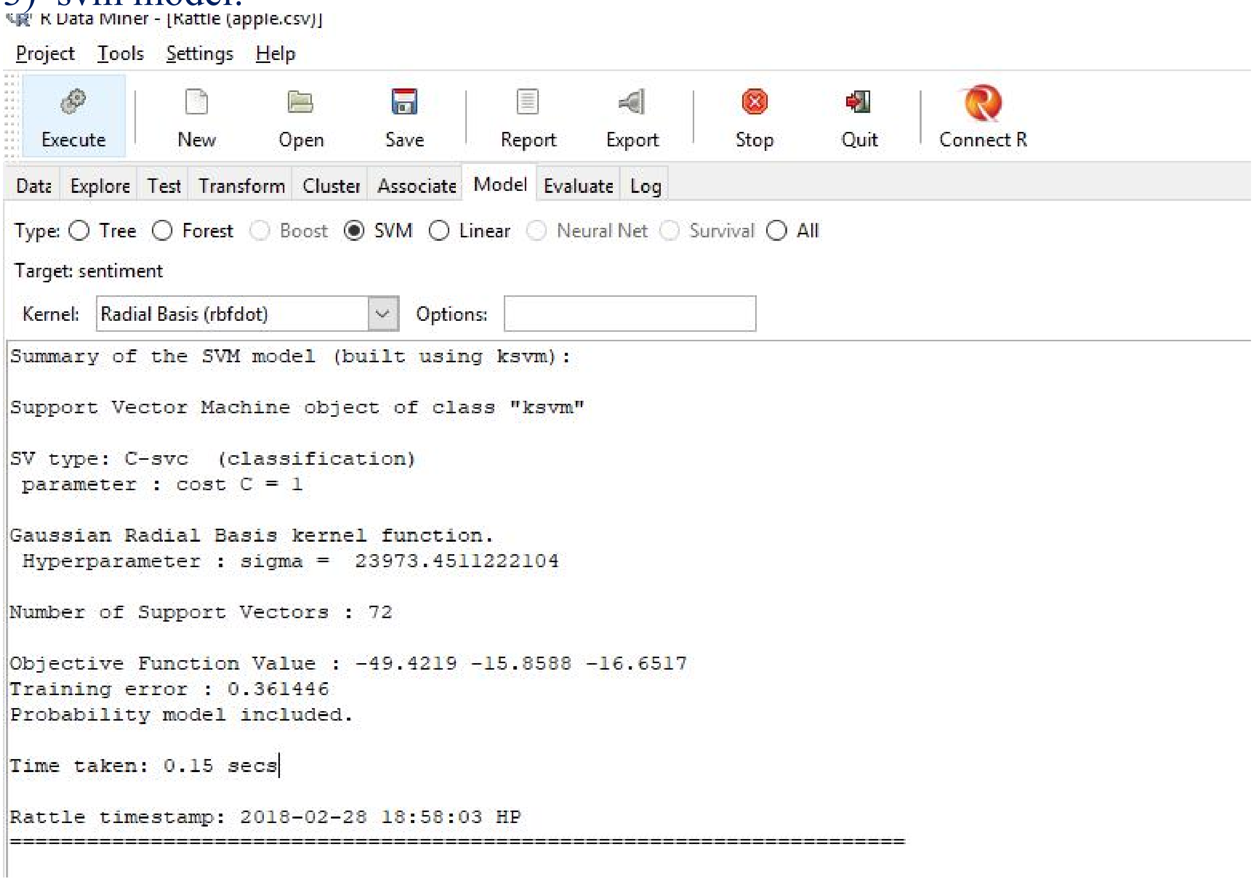
Forest model:-



Error matrix for forest model:-



3) svm model:-



Error matrix for svm model:-

```
Rattle timestamp: 2018-02-28 19:04:28 HP
======================================================================
Error matrix for the SVM model on apple.csv [validate] (counts):

      Predicted
Actual 1  3 5 Error
     1 0  5 0   100
     3 0 14 0     0
     5 0  1 0   100

Error matrix for the SVM model on apple.csv [validate] (proportions):

      Predicted
Actual 1  3 5 Error
     1 0 25 0   100
     3 0 70 0     0
     5 0  5 0   100

Overall error: 30%, Averaged class error: 66.66667%
```

Linear model:-

Data Explore Test Transform Cluster Associate Model Evaluate Log

Type: Tree Forest Boost SVM Linear Neural Net Survival All

Numeric Generalized Poisson Logistic Probit Multinomial Model Builder: multinom

Plot

```
Summary of the Multinomial Regression model (built using multinom):
Call:
multinom(formula = sentiment ~ ., data = crs$dataset[crs$train,
    c(crs$input, crs$target)], trace = FALSE, maxit = 1000)

n=83

Coefficients:
   (Intercept)           id1           id
3  2.17674e-34  1.942703e-16  1.357195e-25
5 -2.10168e-34 -1.875714e-16 -1.310395e-25

Std. Errors:
   (Intercept)          id1           id
3 0.000000e+00 5.489005e+18 3.834686e+09
5 1.219424e-72 1.088315e-54 7.603105e-64

Value/SE (Wald statistics):
    (Intercept)           id1           id
3           Inf  3.539263e-35  3.539260e-35
5 -1.723502e+38 -1.723502e+38 -1.723499e+38

Residual Deviance: 182.3696
AIC: 190.3696
Log likelihood: -91.185 (4 df)
Pseudo R-Square: NA
```

Data Explore Test Transform Cluster Associate Model Evaluate Log

Type: ○ Tree ○ Forest ○ Boost ○ SVM ◉ Linear ○ Neural Net ○ Survival ○ All

○ Numeric ○ Generalized ○ Poisson ○ Logistic ○ Probit ◉ Multinomial Model Builder: multinon

Plot

```
3 0.000000e+00 5.489005e+18 3.834686e+09
5 1.219424e-72 1.088315e-54 7.603105e-64

Value/SE (Wald statistics):
    (Intercept)           idl            id
3           Inf  3.539263e-35  3.539260e-35
5 -1.723502e+38 -1.723502e+38 -1.723499e+38

Residual Deviance: 182.3696
AIC: 190.3696
Log likelihood: -91.185 (4 df)
Pseudo R-Square: NA

==== ANOVA ====
Analysis of Deviance Table (Type II tests)

Response: sentiment
    LR Chisq Df Pr(>Chisq)
idl  -28.281  2          1
id     0.000  2          1
[1] "\n"
Time taken: 0.47 secs

Rattle timestamp: 2018-02-28 19:01:59 HP
======================================================================
```

Error matrix for linear:-

```
Rattle timestamp: 2018-02-28 19:04:28 HP
======================================================================
Error matrix for the Linear model on apple.csv [validate] (counts):

      Predicted
Actual 1  3 5 Error
     1 0  5 0   100
     3 0 14 0     0
     5 0  1 0   100

Error matrix for the Linear model on apple.csv [validate] (proportions):

      Predicted
Actual 1  3 5 Error
     1 0 25 0   100
     3 0 70 0     0
     5 0  5 0   100

Overall error: 30%, Averaged class error: 66.66667%
```

All(modules)

Data | Explore | Test | Transform | Cluster | Associate | Model | Evaluate | Log

Type: ○ Tree ○ Forest ○ Boost ○ SVM ○ Linear ○ Neural Net ○ Survival ◉ All

○ Numeric ○ Generalized ○ Poisson ○ Logistic ○ Probit ◉ Multinomial   Model Builder: multi

Plot

```
Summary of the Multinomial Regression model (built using multinom):
Call:
multinom(formula = sentiment ~ ., data = crs$dataset[crs$train,
    c(crs$input, crs$target)], trace = FALSE, maxit = 1000)

n=83

Coefficients:
   (Intercept)           idl            id
3  2.17674e-34  1.942703e-16  1.357195e-25
5 -2.10168e-34 -1.875714e-16 -1.310395e-25

Std. Errors:
   (Intercept)          idl           id
3 0.000000e+00 5.489005e+18 3.834686e+09
5 1.219424e-72 1.088315e-54 7.603105e-64

Value/SE (Wald statistics):
   (Intercept)           idl            id
3          Inf  3.539263e-35  3.539260e-35
5 -1.723502e+38 -1.723502e+38 -1.723499e+38

Residual Deviance: 182.3696
AIC: 190.3696
Log likelihood: -91.185 (4 df)
Pseudo R-Square: NA
```

Data | Explore | Test | Transform | Cluster | Associate | Model | Evaluate | Log

Type: ○ Tree ○ Forest ○ Boost ○ SVM ○ Linear ○ Neural Net ○ Survival ◉ All

○ Numeric ○ Generalized ○ Poisson ○ Logistic ○ Probit ◉ Multinomial   Model Builder: multinom

Plot

```
3 0.000000e+00 5.489005e+18 3.834686e+09
5 1.219424e-72 1.088315e-54 7.603105e-64

Value/SE (Wald statistics):
   (Intercept)           idl            id
3          Inf  3.539263e-35  3.539260e-35
5 -1.723502e+38 -1.723502e+38 -1.723499e+38

Residual Deviance: 182.3696
AIC: 190.3696
Log likelihood: -91.185 (4 df)
Pseudo R-Square: NA

==== ANOVA ====
Analysis of Deviance Table (Type II tests)

Response: sentiment
    LR Chisq Df Pr(>Chisq)
idl  -28.281  2          1
id     0.000  2          1
[1] "\n"
Time taken: 0.03 secs

Rattle timestamp: 2018-02-28 19:03:20 HP
======================================================================
```

Error rate for the models:-

model	Overall error matrix	error
Linear model	30%	0.521
SVM model	30%	0.36144
Forest model	30 %	0.629
Tree model	35%	0.433

Implementation:-

Tools:- Rstudio

Code:-

```
apple <- read.csv(file.choose(), header = T)
str(apple)
```

```
> # Read file
> apple <- read.csv(file.choose(), header = T)
> str(apple)
'data.frame':	1000 obs. of  18 variables:
 $ text         : Factor w/ 629 levels "#Apple #earnings: How long will #iPhone sales
be on â€˜pauseâ€™? $AAPL  #iPhone8 #Retail #applenews #stocks #Fi"| __truncated__,..:
515 515 395 542 17 479 479 525 527 499 ...
 $ favorited    : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ favoriteCount: int  0 0 0 0 0 0 0 0 0 0 ...
 $ replyToSN    : Factor w/ 36 levels "AdamBuschbacher",..: NA NA NA NA NA NA NA NA NA
 NA ...
 $ created      : Factor w/ 115 levels "8/1/2017 18:37",..: 115 115 115 115 115 115 11
5 115 115 115 ...
 $ truncated    : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ replyToSID   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ id1          : num  8.92e+17 8.92e+17 8.92e+17 8.92e+17 8.92e+17 ...
 $ replyToUID   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ statusSource : Factor w/ 51 levels "<a href=http://127.0.0.1:3000/ rel=nofollow>Twi
tter tweets 111</a>",..: 14 14 7 13 7 13 14 14 11 11 ...
 $ screenName   : Factor w/ 736 levels "__v4gue__","_davidelman",..: 368 423 82 411 39
5 397 462 173 713 367 ...
 $ retweetCount : int  3 3 0 85 0 30 30 9 10 1 ...
 $ isRetweet    : logi  TRUE TRUE FALSE TRUE FALSE TRUE ...
 $ retweeted    : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ longitude    : logi  NA NA NA NA NA NA ...
 $ latitude     : logi  NA NA NA NA NA NA ...
 $ id           : int  623499300 623499301 623499302 623499303 623499304 623499305 623
499306 623499307 623499308 623499309 ...
 $ sentiment    : int  3 3 3 3 1 3 3 3 1 1 ...
> 
```

```
library(tm)
corpus <- iconv(apple$text, to='UTF-8', sub = "byte")
```

```
corpus <- Corpus(VectorSource(corpus))
inspect(corpus[1:10])
```

```
> library(tm)
> corpus <- iconv(apple$text, to='UTF-8', sub = "byte")
> corpus <- Corpus(VectorSource(corpus))
> inspect(corpus[1:10])
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 10

 [1] RT @option_snipper: $AAPL beat on both eps and revenues. SEES 4Q REV. $49B-$52B,
EST. $49.1B https://t.co/hfHXqj0IOB
 [2] RT @option_snipper: $AAPL beat on both eps and revenues. SEES 4Q REV. $49B-$52B,
EST. $49.1B https://t.co/hfHXqj0IOB
 [3] Let's see this break all timers. $AAPL 156.89

 [4] RT @SylvaCap: Things might get ugly for $aapl with the iphone delay. With $aapl d
own that means almost all of the FANG stocks were down posâ€¦
 [5] $AAPL - wow! This was supposed to be a throw-away quarter and AAPL beats by over
500 million in revenue! Trillion dollar company by 2018!
 [6] RT @CNBCnow: EARNINGS: Apple Q3 EPS $1.67 vs. $1.57 Est.; Q3 Revs. $45.4B vs. $44
.89B Est. â€¢ $AAPL https://t.co/UzI8Uh9GJI https://t.co/WzXâ€¦
 [7] RT @CNBCnow: EARNINGS: Apple Q3 EPS $1.67 vs. $1.57 Est.; Q3 Revs. $45.4B vs. $44
.89B Est. â€¢ $AAPL https://t.co/UzI8Uh9GJI https://t.co/WzXâ€¦
 [8] RT @Selerity: #BREAKING: Apple $AAPL Q3 Earnings Per Share (EPS), $1.67 vs. $1.57
 expected
 [9] RT @Selerity: #BREAKING: Apple $AAPL Q3 Revenue, $45.41B vs. $44.9B expected

[10] RT @JackWangCFA: #Apple @apple $aapl #earnings #RealTime #BREAKING Rev $45.4B &am
p; EPS of $1.67 vs #street #consensus #estimate $44.9B; EPS $1â€¦
> 
```

```
corpus <- tm_map(corpus, tolower)

corpus <- tm_map(corpus, removePunctuation)

corpus <- tm_map(corpus, removeNumbers)

cleanset <- tm_map(corpus, removeWords, stopwords('english'))

removeURL <- function(x) gsub('http[[:alnum:]]*', '', x)
cleanset <- tm_map(cleanset, content_transformer(removeURL))
inspect(cleanset[1:5])
```

```
> corpus <- tm_map(corpus, tolower)
> corpus <- tm_map(corpus, removePunctuation)
> corpus <- tm_map(corpus, removeNumbers)
> cleanset <- tm_map(corpus, removeWords, stopwords('english'))
> removeURL <- function(x) gsub('http[[:alnum:]]*', '', x)
> cleanset <- tm_map(cleanset, content_transformer(removeURL))
> inspect(cleanset[1:5])
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 5

 [1] rt optionsnipper aapl beat   eps  revenues sees q rev bb est b

 [2] rt optionsnipper aapl beat   eps  revenues sees q rev bb est b

 [3] lets see  break  timers aapl

 [4] rt sylvacap things might get ugly  aapl   iphone delay  aapl   means almost    fan
g stocks   posâ€¦
 [5] aapl  wow   supposed    throwaway quarter  aapl beats    million  revenue trillion
 dollar company
> 
```

tdm <- TermDocumentMatrix(cleanset)
tdm

```
> tdm <- TermDocumentMatrix(cleanset)
> tdm
<<TermDocumentMatrix (terms: 1654, documents: 1000)>>
Non-/sparse entries: 9324/1644676
Sparsity           : 99%
Maximal term length: 22
Weighting          : term frequency (tf)
> 
```

tdm <- as.matrix(tdm)
tdm[1:10, 1:20]

```
> tdm <- as.matrix(tdm)
> tdm[1:10, 1:20]
               Docs
Terms           1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
  aapl          1 1 1 2 2 1 1 1 1  1  1  1  1  1  2  1  1  1  1  1
  beat          1 1 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  1  0
  eps           1 1 0 0 0 1 1 1 0  2  0  0  0  1  0  0  1  1  0  0
  est           1 1 0 0 0 2 2 0 0  0  0  0  0  2  0  0  2  2  0  0
  optionsnipper 1 1 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0
  rev           1 1 0 0 0 0 0 0 0  1  0  0  1  0  0  0  0  0  0  0
  revenues      1 1 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0
  sees          1 1 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0
  break         0 0 1 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0
  lets          0 0 1 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0
> 
```

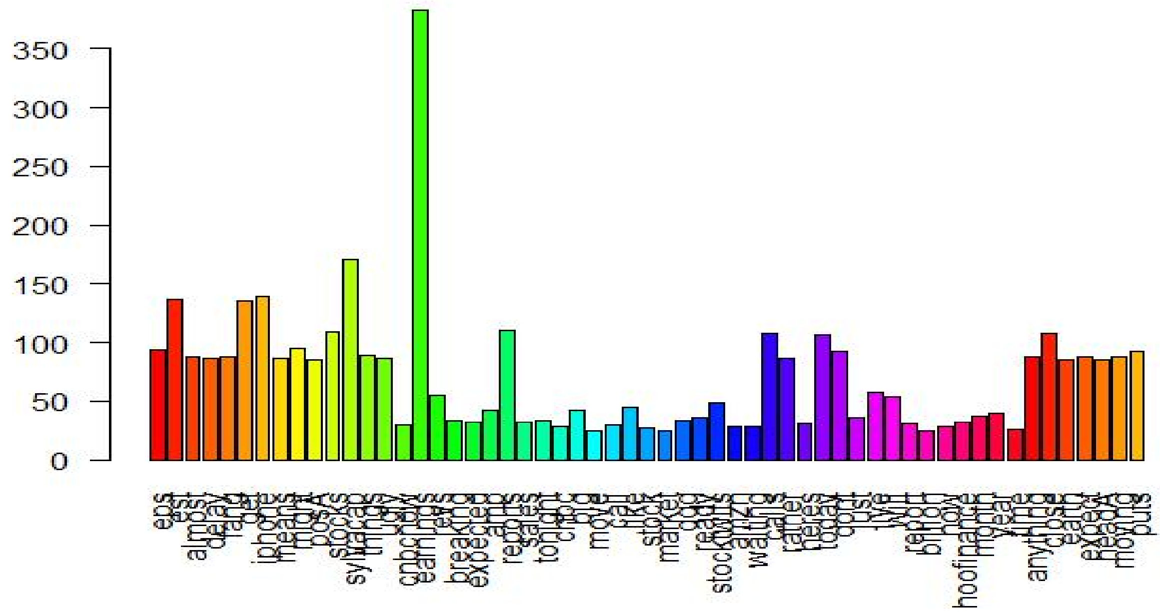
cleanset <- tm_map(cleanset, removeWords, c('aapl', 'apple'))
cleanset <- tm_map(cleanset, stripWhitespace)
inspect(cleanset[1:5])

tdm <- TermDocumentMatrix(cleanset)

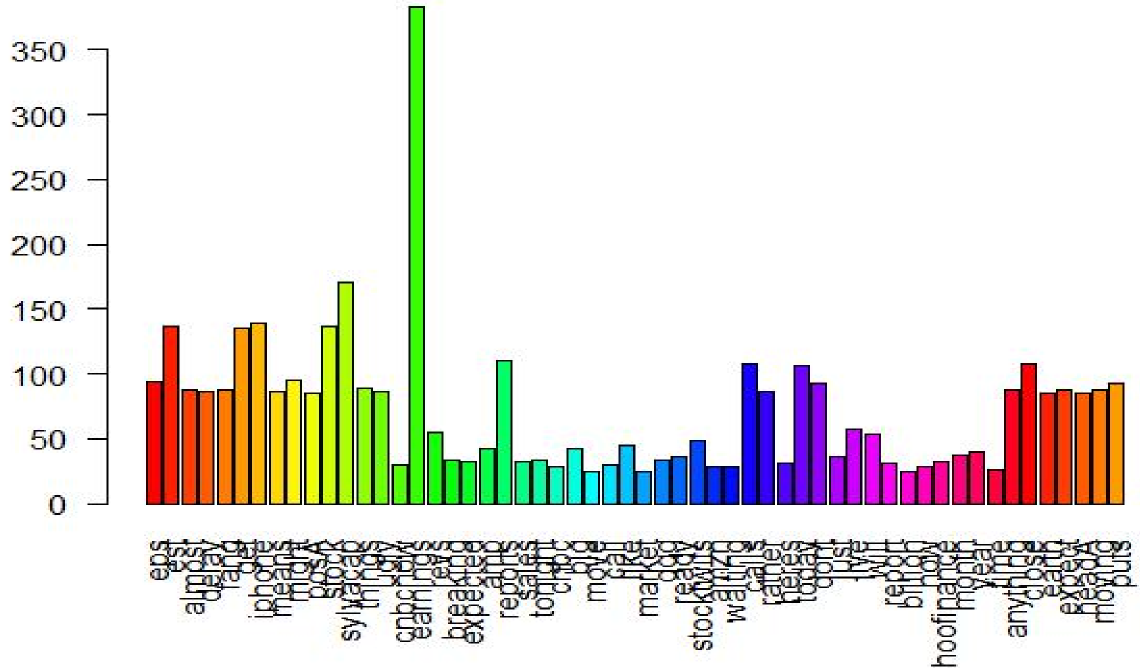
tdm
tdm <- as.matrix(tdm)
tdm[1:10, 1:20]

```
> cleanset <- tm_map(cleanset, removeWords, c('aapl', 'apple'))
> tdm <- TermDocumentMatrix(cleanset)
> tdm
<<TermDocumentMatrix (terms: 1651, documents: 1000)>>
Non-/sparse entries: 8131/1642869
Sparsity           : 100%
Maximal term length: 22
Weighting          : term frequency (tf)
> tdm <- as.matrix(tdm)
> tdm[1:10, 1:20]
               Docs
Terms           1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
  beat          1 1 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  1  0
  eps           1 1 0 0 0 1 1 1 0  2  0  0  0  1  0  0  1  1  0  0
  est           1 1 0 0 0 2 2 0 0  0  0  0  0  2  0  0  2  2  0  0
  optionsnipper 1 1 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0
  rev           1 1 0 0 0 0 0 0 0  1  0  0  1  0  0  0  0  0  0  0
  revenues      1 1 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0
  sees          1 1 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0
  break         0 0 1 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0
  lets          0 0 1 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0
  see           0 0 1 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0
> 
```

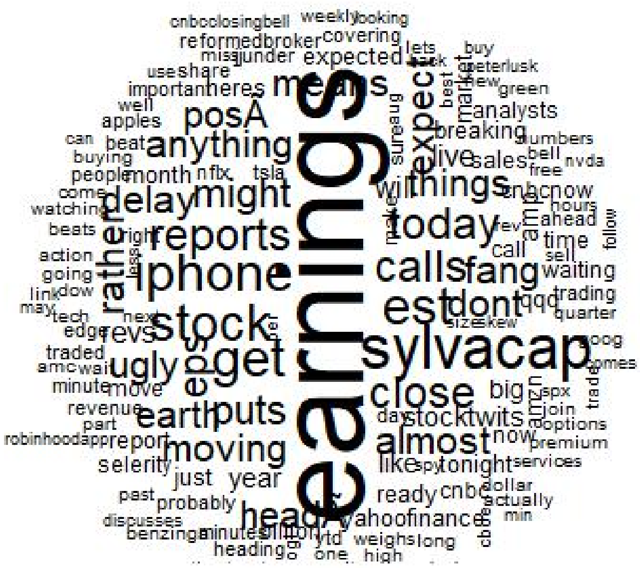
w <- rowSums(tdm)
w <- subset(w, w>=25)
barplot(w, las = 2, col = rainbow(50))



```
cleanset <- tm_map(cleanset, gsub, pattern = 'stocks', replacement = 'stock')
```




```
library(wordcloud)
w <- sort(rowSums(tdm), decreasing = TRUE)
set.seed(222)
wordcloud(words = names(w),
          freq = w,
          max.words = 150,
          random.order = F)
```



```
library(wordcloud2)
w <- data.frame(names(w), w)
colnames(w) <- c('word', 'freq')
wordcloud2(w,
                size = 0.7,
                shape = 'triangle')
```


```
library(syuzhet)
library(lubridate)
library(ggplot2)
library(scales)
library(reshape2)
library(dplyr)

apple <- read.csv(file.choose(), header = T)
tweets <- iconv(apple$text, to ='UTF-8',sub="byte")
s <- get_nrc_sentiment(tweets)
head(s)
```

```
> library(lubridate)
> library(ggplot2)
> library(scales)
> library(reshape2)
> library(dplyr)
> apple <- read.csv(file.choose(), header = T)
> tweets <- iconv(apple$text, to ='UTF-8',sub="byte")
> s <- get_nrc_sentiment(tweets)
> head(s)
  anger anticipation disgust fear joy sadness surprise trust negative positive
1     0            0       0    0   0       0        0     0        0        1
2     0            0       0    0   0       0        0     0        0        1
3     0            0       0    0   0       0        1     0        0        0
4     1            0       2    2   0       1        0     0        3        0
5     0            0       0    0   0       0        0     0        0        0
6     0            0       0    0   0       0        0     0        0        0
> |
```

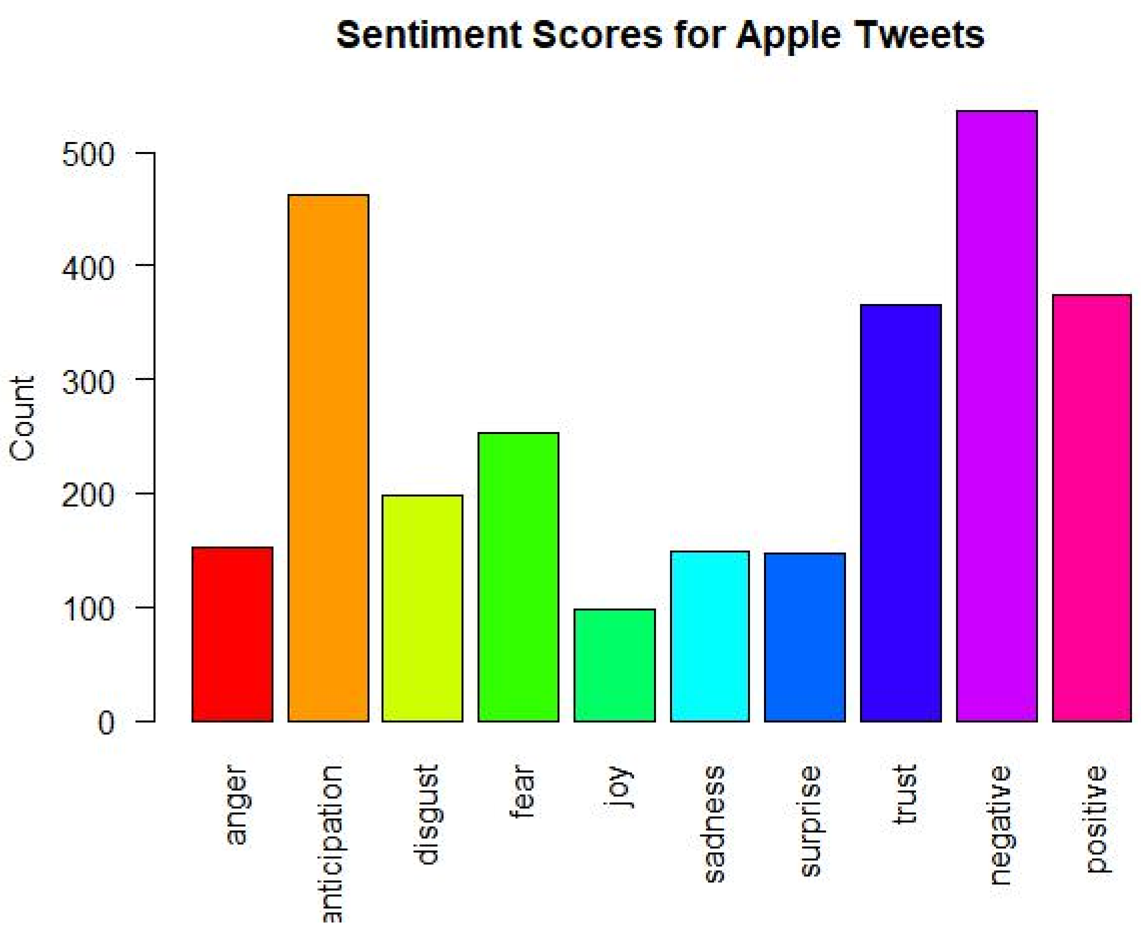
tweets[4]

```
> tweets[4]
[1] "RT @SylvaCap: Things might get ugly for $aapl with the iphone delay. With $aapl d
own that means almost all of the FANG stocks were down posâ€¦"
> |
```

get_nrc_sentiment('ugly')

```
> get_nrc_sentiment('ugly')
  anger anticipation disgust fear joy sadness surprise trust negative positive
1     0            0       1    0   0       0        0     0        1        0
> |
```

barplot(colSums(s),las = 2,col = rainbow(10),ylab = 'Count',main = 'Sentiment Scores for Apple Tweets')



Conclusion:-

Twitter data analysis we can analysis the users tweets using the sentimental analysis Through these sentimental analysis we get the how public are positively, negatively,or nutral with the perticular person. Now a days we are analysis the politics who will win the election. We can do the centimentina analysis the products reviews wither products are positive openion and now a days most of the companys using these methods where they are focusing for the target users to devop their business