Assignment 3 - Text Mining and Sentiment Analysis

Advik, Prerna, Prakhar

4/10/2021

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──
```

```
## ✓ ggplot2 3.3.3     ✓ purrr   0.3.4
## ✓ tibble  3.0.6     ✓ dplyr   1.0.4
## ✓ tidyr   1.1.2     ✓ stringr 1.4.0
## ✓ readr   1.4.0     ✓ forcats 0.5.1
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidytext)
library(SnowballC)
library(textstem)
```

```
## Loading required package: koRpus.lang.en
```

```
## Loading required package: koRpus
```

```
## Loading required package: sylly
```

```
## For information on available language packages for 'koRpus', run
##
##   available.koRpus.lang()
##
## and see ?install.koRpus.lang()
```

```
##
## Attaching package: 'koRpus'
```

```
## The following object is masked from 'package:readr':
##
##     tokenize
```

```
library(textdata)
library(rsample)
library(ranger)
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
#load the data
resReviewsData <- read_csv2('yelpRestaurantReviews_sample.csv')
```

```
## ℹ Using ',' as decimal and '.' as grouping mark. Use `read_delim()` for more control.
```

```
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   .default = col_character(),
##   cool = col_double(),
##   date = col_date(format = ""),
##   funny = col_double(),
##   stars = col_double(),
##   useful = col_double(),
##   is_open = col_double(),
##   latitude = col_number(),
##   longitude = col_number(),
##   review_count = col_double()
## )
## ℹ Use `spec()` for the full column specifications.
```

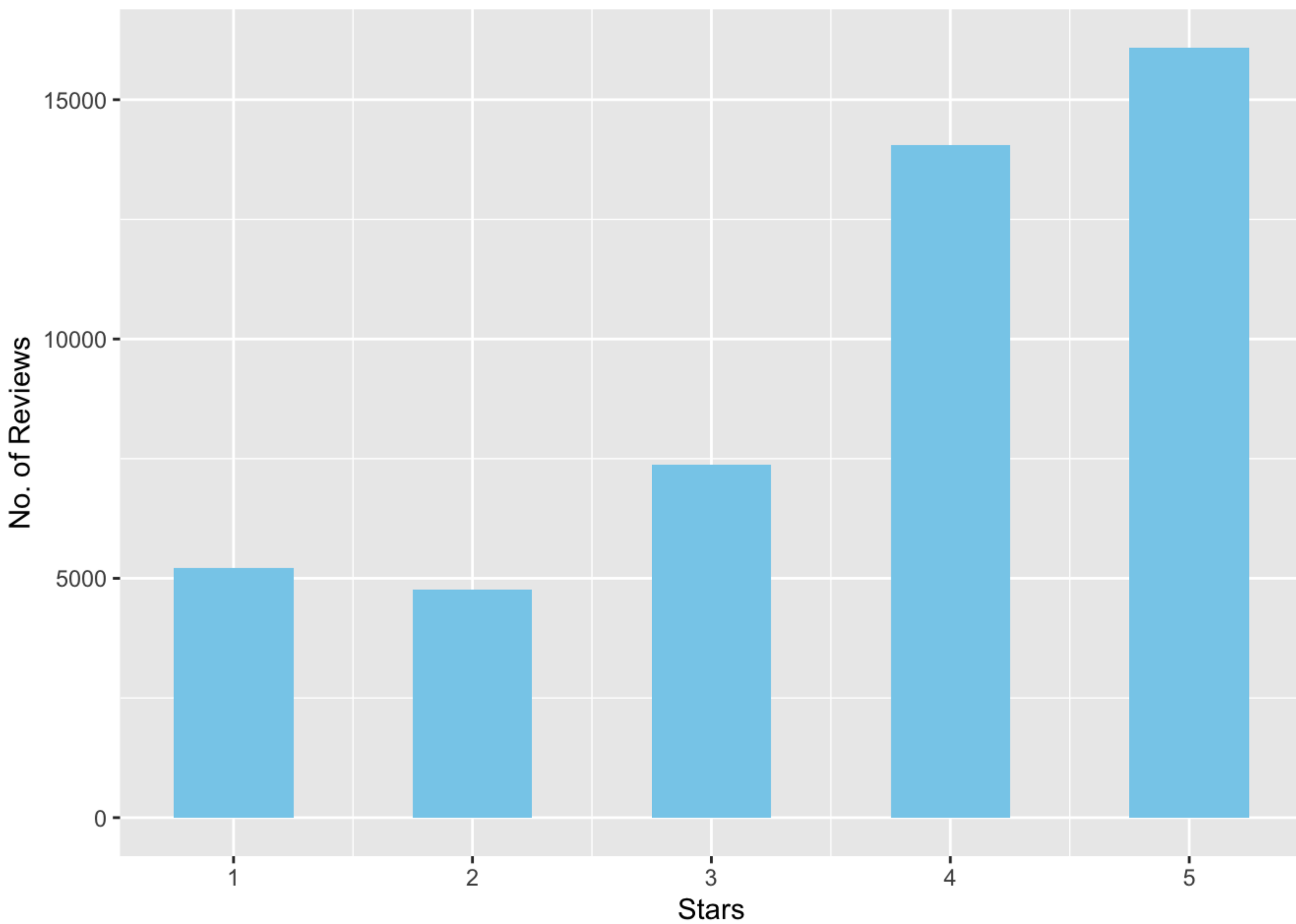
```
#Review distribution across star ratings
resReviewsData %>% group_by(stars) %>% count()
```

stars <dbl>	n <int>
1	5224
2	4757
3	7381
4	14042

stars <dbl>	n <int>
5	16091

5 rows

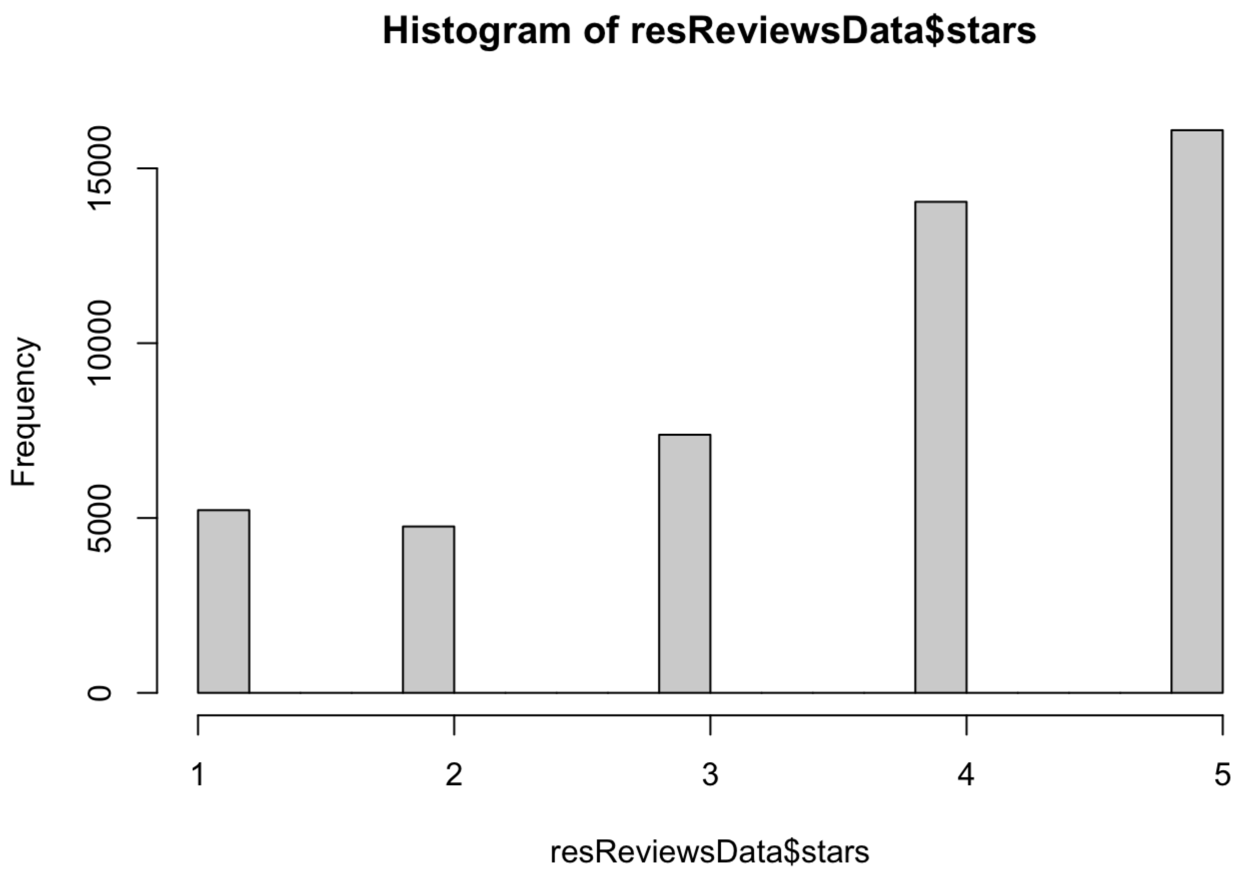
```
#graph to depict the distribution of ratings
ggplot(resReviewsData, aes(x=stars)) + geom_bar(width = 0.5, fill = "sky blue") + xlab(
"Stars") + ylab("No. of Reviews")
```




```
#Review ratings by state
resReviewsData %>%   group_by(state) %>% tally() %>% view()


#Keeping only reviews from 5-digit postal-codes
rrData <- resReviewsData %>% filter(str_detect(postal_code, "^[0-9]{1,5}"))
table <- rrData %>% group_by(postal_code) %>% count()
table <- ungroup(table)
top_postal_code <- table %>% top_n(20)
```

```
## Selecting by n
```

```
#Plotting graphs to see how certain words like cool,funny and useful affect ratings
hist(resReviewsData$stars)
```




```
#tokenize the text of the reviews in the column named 'text'
rrTokens <- rrData %>% unnest_tokens(word, text)
#How many tokens?
rrTokens %>% distinct(word) %>% dim()
```

```
## [1] 70321     1
```

```
#Or we can select just the review_id and the text column
rrTokens <- rrData %>% select(review_id, stars, text ) %>% unnest_tokens(word, text)

#remove stopwords
rrTokens <- rrTokens %>% anti_join(stop_words)
```

```
## Joining, by = "word"
```

```
#compare with earlier - what fraction of tokens were stopwords?
rrTokens %>% distinct(word) %>% dim()
```

```
## [1] 69622     1
```

```
#Remove non alphabetic characters
rrTokens<-rrTokens %>%  filter(!str_detect(word, "[^[:alpha:]]"))
#Dimensions for rrTokens
rrTokens %>% dim()
```

```
## [1] 1249745       3
```

```
#Dimensions for the distinct word tokens
rrTokens %>% distinct(word) %>% dim()
```

```
## [1] 50338     1
```

```
#Stemming
rrTokens<-rrTokens %>%  mutate(word_stem = SnowballC::wordStem(word))
#Dimensions for rrTokens
rrTokens %>% dim()
```

```
## [1] 1249745       4
```

```
#Dimensions for the distinct word_stem tokens
rrTokens %>% distinct(word_stem) %>% dim()
```

```
## [1] 38985     1
```

```
#Lemmatization
rrTokens<-rrTokens %>%  mutate(word_lemma = textstem::lemmatize_words(word))
#Dimensions for rrTokens
rrTokens %>% dim()
```

```
## [1] 1249745       5
```

```
#Dimensions for the distinct word_lemma tokens
rrTokens %>% distinct(word_lemma) %>% dim()
```

```
## [1] 43333     1
```

```
#We move ahead with Lemmatization
rrTokens<-rrTokens %>%  mutate(word = textstem::lemmatize_words(word)) %>% select(-word_
stem, -word_lemma)
#Dimensions for rrTokens
rrTokens %>% dim()
```

```
## [1] 1249745       3
```

```
#Dimensions for the distinct word_stem tokens
rrTokens %>% distinct(word) %>% dim()
```

```
## [1] 43333     1
```

```
#We may want to filter out words with less than 3 characters and those with more than 15
characters
rrTokens<-rrTokens %>% filter(str_length(word)<=3 | str_length(word)<=15)
#Dimensions for rrTokens
rrTokens %>% dim()
```

```
## [1] 1248641       3
```

```
#Dimensions for the distinct word tokens
rrTokens %>% distinct(word) %>% dim()
```

```
## [1] 42478     1
```

```
#count the total occurrences of differet words, & sort by most frequent
rrTokens %>% count(word, sort=TRUE) %>% top_n(10)
```

```
## Selecting by n
```

word <chr>	**n** <int>
food	25514
service	12453
time	12334
restaurant	8838
eat	8366
chicken	7722
love	7505
pizza	6215
nice	6126
fry	5987

1-10 of 10 rows

```
#Are there some words that occur in a large majority of reviews, or which are there in v
ery few reviews?   Let's remove the words which are not present in at least 10 reviews
rareWords <-rrTokens %>% count(word, sort=TRUE) %>% filter(n<10)
#dimension for distinct rare words
rareWords %>% distinct(word) %>% dim()
```

```
## [1] 35303     1
```

```
#remove rare words
rrTokens<-anti_join(rrTokens, rareWords)
```

```
## Joining, by = "word"
```

```
#Dimensions for rrTokens
rrTokens %>% dim()
```

```
## [1] 1176351       3
```

```
#dimension for distinct words after removing rare words
rrTokens %>% distinct(word) %>% dim()
```

```
## [1] 7175    1
```

```
#proportion of word occurrence for different star ratings
ws<-rrTokens %>% group_by(stars) %>% count(word, sort=TRUE)
ws<-ws %>% group_by(stars) %>% mutate(prop=n/sum(n)) %>% arrange(desc(stars, prop))
#proportion of word occurrence wrt different star ratings (top 20 for each)
table2 <- ws %>% group_by(stars) %>% arrange(stars, desc(prop)) %>% top_n(20)
```

```
## Selecting by prop
```

```
#what are the most commonly used words by start rating
ws %>% group_by(stars) %>% arrange(stars, desc(prop)) %>% view()

#to see the top 20 words by star ratings
ws %>% group_by(stars) %>% arrange(stars, desc(prop)) %>% filter(row_number()<=20) %>% v
iew()

xx<- ws %>% group_by(word) %>% summarise(totWS=sum(stars*prop))
#What are the 20 words with highest and lowerst star rating
gtop_20<-xx %>% top_n(20)
```

```
## Selecting by totWS
```

```
xx %>% top_n(20)
```

```
## Selecting by totWS
```

word <chr>	totWS <dbl>
cheese	0.05954374
chicken	0.09888377
delicious	0.07174952
drink	0.06014871
eat	0.10566833
food	0.32452907
friendly	0.06204531
fry	0.07753366
love	0.09699265
menu	0.07230269

1-10 of 20 rows Previous **1** 2 Next

```
xx %>% top_n(-20)
```

```
## Selecting by totWS
```

word <chr>	totWS <dbl>
brazilian	1.037623e-04
bullshit	1.100723e-04
coffe	1.059444e-04
dispute	1.008259e-04
disrespect	1.064397e-04
disrespectful	9.451596e-05
evidently	1.095769e-04
inspector	1.023379e-04
intent	1.095639e-04
laminate	1.059444e-04

1-10 of 20 rows Previous 1 2 Next

```
#make a copy of rrTokens
rrTokens1 <- rrTokens

#calculate tf, idf and tf-idf
rrTokens <- rrTokens %>% group_by(review_id, stars) %>% count(word)
rrTokens <- rrTokens %>% bind_tf_idf(word, review_id, n)

#ungroup rrTokens
rrTokens <- ungroup(rrTokens)
```

```
#take a look at the words in the sentimennt dictionaries
get_sentiments("bing") %>% view()
get_sentiments("nrc") %>% view()
get_sentiments("afinn") %>% view()

######################## BING DICTIONARY #########################
#get the sentiment of words in rrTokens from Bing
rrSenti_bing<- rrTokens %>% inner_join(get_sentiments("bing"), by="word")
#Dimensions for rrSenti_bing
rrSenti_bing %>% dim()
```

```
## [1] 173652      8
```

```
#dimension for distinct words after performing inner join with Bing
rrSenti_bing %>% distinct(word) %>% dim()
```

```
## [1] 1020    1
```

```
#count the total occurence of words
xx<-rrSenti_bing %>% group_by(word, sentiment) %>% summarise(totOcc=sum(n)) %>% arrange
(sentiment, desc(totOcc))
```

```
## `summarise()` has grouped output by 'word'. You can override using the `.groups` argu
ment.
```

```
#word count and word occurence for different sentiment categories
xx1 <- xx %>% group_by(sentiment) %>% summarise(count=n(), sumn=sum(totOcc))

#negate count for negative sentiment words
xx<- xx %>% mutate (totOcc=ifelse(sentiment=="positive", totOcc, -totOcc))

#Ungroup xx
xx<-ungroup(xx)

revSenti_bing <- rrSenti_bing %>% group_by(review_id, stars) %>% summarise(nwords=n(),po
sSum=sum(sentiment=='positive'), negSum=sum(sentiment=='negative'))
```

```
## `summarise()` has grouped output by 'review_id'. You can override using the `.groups`
argument.
```

```
#summarise positive/negative sentiment words proportion per review
revSenti_bing<- revSenti_bing %>% mutate(posProp=posSum/nwords, negProp=negSum/nwords)

#calculate sentiment score
revSenti_bing<- revSenti_bing %>% mutate(sentiScore=posProp-negProp)

#calculate average sentiment score for each rating
bing_star_table <- revSenti_bing %>% group_by(stars) %>% summarise(avgPos=mean(posProp),
avgNeg=mean(negProp), avgSentiSc=mean(sentiScore))


revSenti_bing <- revSenti_bing %>% mutate(hiLo=ifelse(stars<=2,-1, ifelse(stars>=4, 1, 0
)))
revSenti_bing <- revSenti_bing %>% mutate(pred_hiLo=ifelse(sentiScore >0, 1, -1))

#filter out the reviews with 3 stars, and get the confusion matrix for hiLo vs pred_hiLo
final<-revSenti_bing %>% filter(hiLo!=0)
cm <- table(actual=final$hiLo, predicted=final$pred_hiLo)


# ################### NRC Dictionary ####################

#get the sentiment of words in rrTokens from NRC
rrSenti_nrc<-rrTokens %>% inner_join(get_sentiments("nrc"), by="word")

#count the total occurence of words
xx<-rrSenti_nrc %>% group_by(word, sentiment) %>% summarise(totOcc=sum(n)) %>% arrange(s
entiment, desc(totOcc))
```

```
## `summarise()` has grouped output by 'word'. You can override using the `.groups` argu
ment.
```

```
#word count and word occurence for different sentiment categories
xx1 <- xx %>% group_by(sentiment) %>% summarise(count=n(), sumn=sum(totOcc))


#consider {anticipation, joy, positive, surprise, trust} as positive reviews (Positive t
otOcc)
#consider {anger, disgust, fear, negative, sadness} as negative reviews      (Negative t
otOcc)
xx<-xx %>% mutate(totOcc=ifelse(sentiment %in% c('anger', 'disgust', 'fear', 'negative',
'sadness'), -totOcc, ifelse(sentiment %in% c('anticipation', 'joy', 'positive', 'surpris
e', 'trust'), totOcc, 0)))

#classify into only 2 categories (positive and negative) based on totOcc
xx<-xx %>% mutate(posNeg=ifelse(totOcc >0, 'positive', 'negative'))

#Ungroup xx
xx<-ungroup(xx)


#summarise number of positive/negative sentiment words per review
revSenti_nrc <- rrSenti_nrc %>% group_by(review_id, stars) %>% summarise(nwords=n(),posS
um=sum(sentiment %in% c('anticipation', 'joy', 'positive', 'surprise', 'trust')), negSum
=sum(sentiment %in% c('anger', 'disgust', 'fear', 'negative', 'sadness')))
```

```
## `summarise()` has grouped output by 'review_id'. You can override using the `.groups`
argument.
```

```
revSenti_nrc <- revSenti_nrc %>% mutate(hiLo=ifelse(stars<=2,-1, ifelse(stars>=4, 1, 0
 )))

#summarise positive/negative sentiment words proportion per review
revSenti_nrc<- revSenti_nrc %>% mutate(posProp=posSum/nwords, negProp=negSum/nwords)

#calculate sentiment score
revSenti_nrc<- revSenti_nrc %>% mutate(sentiScore=posProp-negProp)
revSenti_nrc <- revSenti_nrc %>% mutate(pred_hiLo=ifelse(sentiScore >0, 1, -1))
final<-revSenti_nrc %>% filter(hiLo!=0)
cm <- table(actual=final$hiLo, predicted=final$pred_hiLo)

# ################### Afinn Dictionary ####################

#get the sentiment of words in rrTokens from Afinn
rrSenti_afinn<- rrTokens %>% inner_join(get_sentiments("afinn"), by="word")

#count the total occurence of words
xx<-rrSenti_afinn %>% group_by(word, value) %>% summarise(totOcc=sum(n)) %>% arrange(val
ue, desc(totOcc))
```

```
## `summarise()` has grouped output by 'word'. You can override using the `.groups` argu
ment.
```

```
#word count and word occurence for different sentiment categories
xx1 <- xx %>% group_by(value) %>% summarise(count=n(), sumn=sum(totOcc))

#negate count for negative sentiment words (based on value)
xx<- xx %>% mutate (totOcc=ifelse(value>0, totOcc, -totOcc))

#classify into only 2 categories (positive and negative) based on totOcc
xx<-xx %>% mutate(posNeg=ifelse(totOcc >0, 'positive', 'negative'))

#Ungroup xx
xx<-ungroup(xx)

#top 25 positive words based on total occurence
afinnPos_25 <- xx %>% top_n(n=25, wt=totOcc)


#summarise number of positive/negative sentiment words per review
revSenti_afinn <- rrSenti_afinn %>% group_by(review_id, stars) %>% summarise(nwords=n(),
posSum=sum(value>0), negSum=sum(value<0))
```

```
## `summarise()` has grouped output by 'review_id'. You can override using the `.groups`
argument.
```

```
#summarise positive/negative sentiment words proportion per review
revSenti_afinn<- revSenti_afinn %>% mutate(posProp=posSum/nwords, negProp=negSum/nwords)

#calculate sentiment score
revSenti_afinn<- revSenti_afinn %>% mutate(sentiScore=posProp-negProp)

revSenti_afinn <- revSenti_afinn %>% mutate(hiLo=ifelse(stars<=2,-1, ifelse(stars>=4, 1,
0 )))
revSenti_afinn <- revSenti_afinn %>% mutate(pred_hiLo=ifelse(sentiScore >0, 1, -1))

#filter out the reviews with 3 stars, and get the confusion matrix for hiLo vs pred_hiLo
final<-revSenti_afinn %>% filter(hiLo!=0)
cm <- table(actual=final$hiLo, predicted=final$pred_hiLo)
```

```
###PART c for all dictionaries ##

# Can we classify reviews on high/low stats based on aggregated sentiment of words in th
e reviews
#we can consider reviews with 1 to 2 stars as positive, and this with 4 to 5 stars as ne
gative

#Compared pos-neg derived from 'stars' VS 3 'dictionary'
remove(xx)
remove(xxnrc)
```

```
## Warning in remove(xxnrc): object 'xxnrc' not found
```

```
remove(xx2)
```

```
## Warning in remove(xx2): object 'xx2' not found
```

```
remove(rrTokens_stem)
```

```
## Warning in remove(rrTokens_stem): object 'rrTokens_stem' not found
```

```
remove(rrTokens_lemm)
```

```
## Warning in remove(rrTokens_lemm): object 'rrTokens_lemm' not found
```

```
memory.limit(size=60000)
```

```
## Warning: 'memory.limit()' is Windows-specific
```

```
## [1] Inf
```

```
#Bing
revSenti_bing <- rrSenti_bing %>% mutate(hiLo=ifelse(stars<=2,-1, ifelse(stars>=4, 1, 0
 )))
revSenti_bing <- revSenti_bing %>% mutate(pred_hiLo=ifelse(sentiment=="positive", 1, -1
))
head(revSenti_bing)
```

review_id <chr>	stars <dbl>	word <chr>	n <int>	tf <dbl>	idf <dbl>	tf_idf <dbl>	sentiment <chr>	hi <c
__-LvVFplBRmVu4o2MQZOw	5	fresh	1	0.04000000	2.264201	0.09056804	positive	
__-LvVFplBRmVu4o2MQZOw	5	friendly	1	0.04000000	2.047289	0.08189158	positive	
__-LvVFplBRmVu4o2MQZOw	5	fun	1	0.04000000	3.527097	0.14108388	positive	
__-LvVFplBRmVu4o2MQZOw	5	ready	1	0.04000000	4.147789	0.16591156	positive	
__27pNlKe_MLYkU4vYR3KA	4	celebrate	1	0.02325581	5.390744	0.12536613	positive	
__27pNlKe_MLYkU4vYR3KA	4	delicious	1	0.02325581	1.988305	0.04623965	positive	

6 rows

```
revSenti_bing <- revSenti_bing %>% drop_na(pred_hiLo)
xx<-revSenti_bing %>% filter(hiLo!=0)
table(actual=xx$hiLo, predicted=xx$pred_hiLo )
```

```
##       predicted
## actual    -1     1
##     -1 21575 13974
##     1  26029 85917
```

```
#NRC
revSenti_nrc <- rrTokens %>% left_join(get_sentiments("nrc"), by="word")
# Positive: "anticipation","joy","positive","trust", "surprise"
# Negative: "anger", "disgust", "fear", "negative", "sadness"
# else NA: 0
revSenti_nrc <- revSenti_nrc %>% mutate(hiLo=ifelse(stars<=2,-1, ifelse(stars>=4, 1, 0
 )))
revSenti_nrc <- revSenti_nrc %>% drop_na(sentiment)
revSenti_nrc <- revSenti_nrc %>%mutate(pred_hiLo=ifelse(sentiment %in% c('anger', 'disgu
st','fear', 'sadness','negative'), -1, ifelse(sentiment %in% c('positive', 'joy', 'antic
ipation', 'trust'), 1, 0)))

xx<-revSenti_nrc %>% filter(hiLo!=0)
xx<-xx %>% filter(pred_hiLo!=0)
table(actual=xx$hiLo, predicted=xx$pred_hiLo)
```

```
##       predicted
## actual     -1      1
##     -1  62252  81480
##     1   89703 299686
```

```
#AFINN

#AFINN carries a numeric value for positive/negative sentiment -- how would you use thes
e

#with AFINN dictionary words....following similar steps as above, but noting that AFINN
# assigns negative to positive sentiment value for words matching the dictionary
rrSenti_afinn<- rrTokens %>% inner_join(get_sentiments("afinn"), by="word")


revSenti_afinnx <- rrSenti_afinn %>% group_by(review_id, stars) %>% dplyr::summarise(nwo
rds=n(), sentiSum =sum(value))
```

```
## `summarise()` has grouped output by 'review_id'. You can override using the `.groups`
argument.
```

```
revSenti_afinnW <- rrSenti_afinn %>% group_by(word) %>% dplyr::summarise(nwords=n(), sen
tiSum =sum(value)) %>% arrange(sentiSum)

xx <- revSenti_afinnW
head(xx,10)
```

word <chr>	nwords <int>	sentiSum <dbl>
bad	3424	-10272
die	1472	-4416
disappoint	2188	-4376
leave	2195	-2195
wrong	1048	-2096
horrible	663	-1989
stop	1986	-1986
terrible	608	-1824
miss	869	-1738
lack	787	-1574

1-10 of 10 rows

```
xx<-ungroup(xx)
top_n(xx, 10)
```

```
## Selecting by sentiSum
```

word <chr>	nwords <int>	sentiSum <dbl>
enjoy	2674	5348
happy	1941	5823
super	1962	5886
recommend	3102	6204
excellent	2315	6945
amaze	3500	7000
friendly	4559	9118
awesome	2333	9332
nice	5059	15177
love	5922	17766

1-10 of 10 rows

```
top_n(xx, -10)
```

```
## Selecting by sentiSum
```

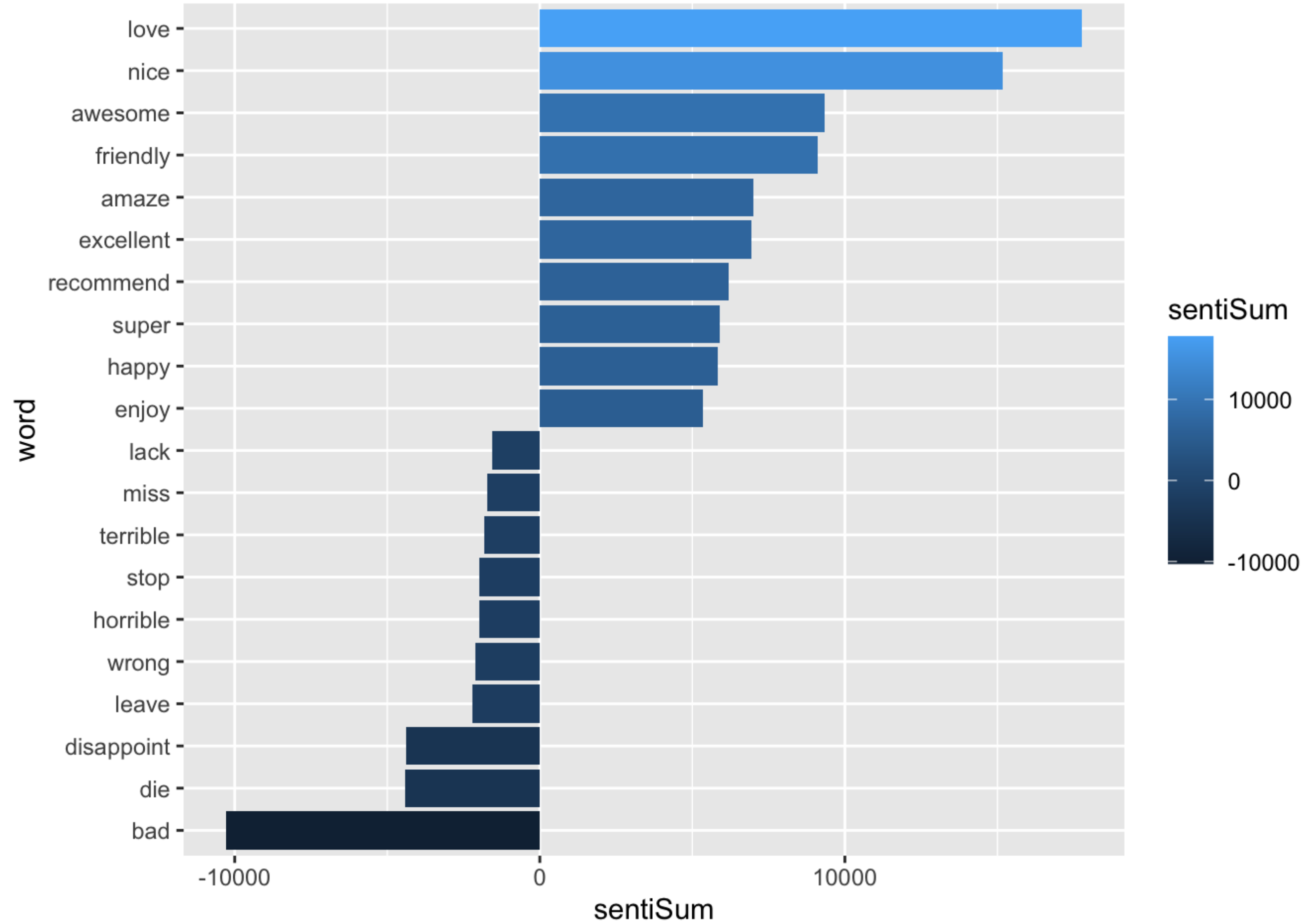
word <chr>	nwords <int>	sentiSum <dbl>
bad	3424	-10272
die	1472	-4416
disappoint	2188	-4376
leave	2195	-2195
wrong	1048	-2096
horrible	663	-1989
stop	1986	-1986
terrible	608	-1824
miss	869	-1738
lack	787	-1574

1-10 of 10 rows

```
rbind(top_n(xx, 10), top_n(xx, -10)) %>% mutate(word=reorder(word, sentiSum)) %>% ggplot
(aes(word, sentiSum, fill=sentiSum)) +geom_col()+coord_flip()
```

```
## Selecting by sentiSum
```

```
## Selecting by sentiSum
```




```
revSenti_afinnx %>% group_by(stars) %>% summarise(avgLen=mean(nwords), avgSenti=mean(sen
tiSum))
```

	stars <dbl>	avgLen <dbl>	avgSenti <dbl>
1	1	4.084541	-2.4565217
2	2	4.363893	0.6549451
3	3	4.382163	3.1361482
4	4	4.291191	5.5288607
5	5	4.005958	6.4692728

5 rows

```
revSenti_afinnW <- rrSenti_afinn %>% group_by(word) %>% dplyr::summarise(nwords=n(), sen
tiSum =sum(value)) %>% arrange(sentiSum)
head(revSenti_afinnx)
```

review_id <chr>	stars <dbl>	nwords <int>	sentiSum <dbl>
__-LvVFplBRmVu4o2MQZOw	5	4	6

review_id	stars	nwords	sentiSum
<chr>	<dbl>	<int>	<dbl>
__27pNlKe_MLYkU4vYR3KA	4	4	6
__2WHmffQO32tNPCLLnaBA	2	1	-3
__E3dqkFaXrzs-bneGDaoA	3	3	7
__Jel-YJhj7iW7CPUnwLOw	5	1	1
__JSApiBsYYBQG-iTQP6cw	5	1	2

6 rows

```
revSenti_afinn <- revSenti_afinnx %>% mutate(hiLo=ifelse(stars<=2,-1, ifelse(stars>=4, 1
, 0 )))
# pred_hiLo is mapping sentiSum as positive and negative
revSenti_afinn <- revSenti_afinn %>% mutate(pred_hiLo=ifelse(sentiSum >0, 1, -1))
#filter out the reviews with 3 stars, and get the confusion matrix for hiLo vs pred_hiLo
xx<-revSenti_afinn %>% filter(hiLo!=0)
table(actual=xx$hiLo, predicted=xx$pred_hiLo )
```

```
##       predicted
## actual    -1     1
##     -1  4476  2435
##     1   2727 18812
```

```
###########################################################################################
############################################



#Can we learn a model to predict hiLo ratings, from words in reviews
#considering only those words which match a sentiment dictionary (for eg.  bing)
#use pivot_wider to convert to a dtm form where each row is for a review and columns cor
respond to words
# (https://tidyr.tidyverse.org/reference/pivot_wider.html)
#revDTM_sentiBing <- rrSenti_bing %>%  pivot_wider(id_cols = review_id, names_from = wor
d, values_from = tf_idf)
#Or, since we want to keep the stars column

dim(rrSenti_bing)
```

```
## [1] 173652      8
```

```
names(rrSenti_bing)
```

```
## [1] "review_id" "stars"     "word"      "n"         "tf"        "idf"
## [7] "tf_idf"    "sentiment"
```

```
sum(is.na(rrSenti_bing$sentiment))
```

```
## [1] 0
```

```
revDTM_sentiBing <- rrSenti_bing %>%pivot_wider(id_cols = c(review_id,stars), names_from
= word, values_from = tf_idf)  %>% ungroup()
#Note the ungroup() at the end -- this is IMPORTANT;  we have grouped based on (review_i
d, stars), and
#this grouping is retained by default, and can cause problems in the later steps
dim(revDTM_sentiBing)
```

```
## [1] 33817  1022
```

```
view(head(revDTM_sentiBing, 10))

#filter out the reviews with stars=3, and calculate hiLo sentiment 'class'
revDTM_sentiBing <- revDTM_sentiBing %>% filter(stars!=3) %>% mutate(hiLo=ifelse(stars<=
2, -1, 1)) %>% select(-stars)
dim(revDTM_sentiBing)
```

```
## [1] 29031  1022
```

```
head(revDTM_sentiBing)
```

review_id <chr>	fresh <dbl>	friendly <dbl>	fun <dbl>	ready <dbl>	celebrate <dbl>	deliciou <dbl>
__-LvVFplBRmVu4o2MQZOw	0.09056804	0.08189158	0.1410839	0.1659116	NA	NA
__27pNlKe_MLYkU4vYR3KA	NA	NA	NA	NA	0.1253661	0.04623965
__2WHmffQO32tNPCLLnaBA	NA	NA	NA	NA	NA	NA
__Jel-YJhj7iW7CPUnwLOw	0.17416931	NA	NA	NA	NA	NA
__JSApiBsYYBQG-iTQP6cw	NA	NA	NA	NA	NA	0.16569207
__VZ1owFav0mn9OE1V_Q6g	NA	NA	NA	NA	NA	NA

6 rows | 1-8 of 1022 columns

```
#how many review with 1, -1  'class'
revDTM_sentiBing %>% group_by(hiLo) %>% tally()
```

	hiLo <dbl>	n <int>
1	-1	7052

	hiLo <dbl>	n <int>
2	1	21979

2 rows

```
############################################ Part C ######################################
#########################################
```

Bing Dictionary

```
#create Document Term Matrix
revDTM_sentiBing <- rrSenti_bing %>%  pivot_wider(id_cols = c(review_id,stars), names_fr
om = word, values_from = tf_idf)  %>% ungroup()

#filter out the reviews with stars=3
#calculate hiLo sentiment(1 is assigned to 4 and 5/-1 is assigned to 1 and 2)
revDTM_sentiBing <- revDTM_sentiBing %>% filter(stars!=3) %>% mutate(hiLo=ifelse(stars<=
2, -1, 1)) %>% select(-stars)

#replace all NAs with zero
revDTM_sentiBing<-revDTM_sentiBing %>% replace(., is.na(.), 0)

#convert hiLo from num to factor
revDTM_sentiBing$hiLo<- as.factor(revDTM_sentiBing$hiLo)


#no of reviews with 1, -1 class
Bing_hiLo_count <- revDTM_sentiBing %>% group_by(hiLo) %>% tally()

set.seed(1234)

#split the data into training and test dataset (50:50)
revDTM_sentiBing_split<- initial_split(revDTM_sentiBing, 0.5)
revDTM_sentiBing_trn  <- training(revDTM_sentiBing_split)
revDTM_sentiBing_tst  <- testing(revDTM_sentiBing_split)
```

#RF Model - 1

```
rfModel1<-ranger(dependent.variable.name = "hiLo", data=revDTM_sentiBing_trn %>% select
(-review_id), num.trees = 500, importance='permutation', probability = TRUE)
```

```
## Growing trees.. Progress: 83%. Estimated remaining time: 6 seconds.
## Computing permutation importance.. Progress: 4%. Estimated remaining time: 11 minute
s, 47 seconds.
## Computing permutation importance.. Progress: 9%. Estimated remaining time: 10 minute
s, 26 seconds.
## Computing permutation importance.. Progress: 13%. Estimated remaining time: 10 minute
s, 24 seconds.
## Computing permutation importance.. Progress: 18%. Estimated remaining time: 9 minute
s, 34 seconds.
## Computing permutation importance.. Progress: 23%. Estimated remaining time: 8 minute
s, 58 seconds.
## Computing permutation importance.. Progress: 27%. Estimated remaining time: 8 minute
s, 26 seconds.
## Computing permutation importance.. Progress: 32%. Estimated remaining time: 7 minute
s, 51 seconds.
## Computing permutation importance.. Progress: 37%. Estimated remaining time: 7 minute
s, 17 seconds.
## Computing permutation importance.. Progress: 42%. Estimated remaining time: 6 minute
s, 42 seconds.
## Computing permutation importance.. Progress: 47%. Estimated remaining time: 6 minute
s, 7 seconds.
## Computing permutation importance.. Progress: 51%. Estimated remaining time: 5 minute
s, 35 seconds.
## Computing permutation importance.. Progress: 56%. Estimated remaining time: 5 minute
s, 1 seconds.
## Computing permutation importance.. Progress: 61%. Estimated remaining time: 4 minute
s, 28 seconds.
## Computing permutation importance.. Progress: 66%. Estimated remaining time: 3 minute
s, 55 seconds.
## Computing permutation importance.. Progress: 71%. Estimated remaining time: 3 minute
s, 23 seconds.
## Computing permutation importance.. Progress: 75%. Estimated remaining time: 2 minute
s, 51 seconds.
## Computing permutation importance.. Progress: 80%. Estimated remaining time: 2 minute
s, 16 seconds.
## Computing permutation importance.. Progress: 85%. Estimated remaining time: 1 minute,
43 seconds.
## Computing permutation importance.. Progress: 90%. Estimated remaining time: 1 minute,
7 seconds.
## Computing permutation importance.. Progress: 95%. Estimated remaining time: 31 second
s.
```

```
#Make predictions from the model on trn and test dataset
revSentiBing_predTrn<- predict(rfModel1, revDTM_sentiBing_trn %>% select(-review_id))
revSentiBing_predTst<- predict(rfModel1, revDTM_sentiBing_tst %>% select(-review_id))

#find the optimal TH
rocTrn <- roc(revDTM_sentiBing_trn$hiLo, revSentiBing_predTrn$predictions[,2], levels=c
(-1, 1))
```

```
## Setting direction: controls < cases
```

```
rocTst <- roc(revDTM_sentiBing_tst$hiLo, revSentiBing_predTst$predictions[,2], levels=c
(-1, 1))
```

```
## Setting direction: controls < cases
```

```
#Best threshold from ROC analyses
bThr<-coords(rocTrn, "best", ret="threshold", transpose = FALSE)
#table(actual=revDTM_sentiBing_trn$hiLo, preds=revSentiBing_predTrn[,2]>bThr)

#Confusion Matrix at bThr for Trn and Tst dataset
a <- table(actual=revDTM_sentiBing_trn$hiLo, preds=revSentiBing_predTrn$predictions[,2]>
0.5)
b <- table(actual=revDTM_sentiBing_tst$hiLo, preds=revSentiBing_predTst$predictions[,2]>
0.5)

auc(as.numeric(revDTM_sentiBing_trn$hiLo), revSentiBing_predTrn$predictions[,2])
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.9889
```

```
auc(as.numeric(revDTM_sentiBing_tst$hiLo), revSentiBing_predTst$predictions[,2])
```

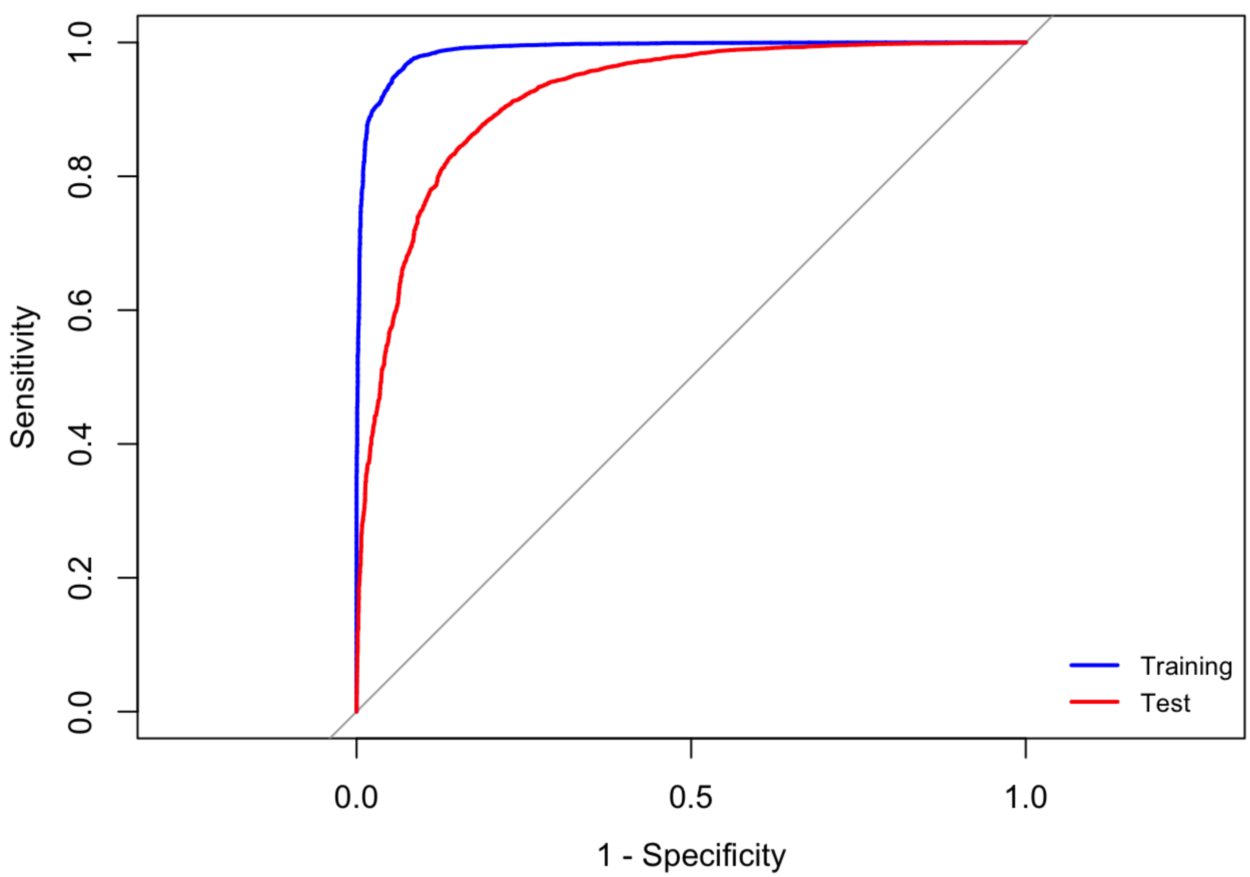
```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.9192
```

```
#which variables are important
importance(rfModel1) %>% view()

rfModel1
```

```
## Ranger result
##
## Call:
##  ranger(dependent.variable.name = "hiLo", data = revDTM_sentiBing_trn %>%      select
(-review_id), num.trees = 500, importance = "permutation",      probability = TRUE)
##
## Type:                             Probability estimation
## Number of trees:                  500
## Sample size:                      14516
## Number of independent variables:  1020
## Mtry:                             31
## Target node size:                 10
## Variable importance mode:         permutation
## Splitrule:                        gini
## OOB prediction error (Brier s.):  0.0910294
```

```r
library(pROC)
#rocTrn <- roc(revDTM_sentiBing_trn$hiLo, revSentiBing_predTrn[,2], levels=c(-1, 1))
#rocTst <- roc(revDTM_sentiBing_tst$hiLo, revSentiBing_predTst[,2], levels=c(-1, 1))

plot.roc(rocTrn, col='blue', legacy.axes = TRUE)
plot.roc(rocTst, col='red', add=TRUE)
legend("bottomright", legend=c("Training", "Test"),
        col=c("blue", "red"), lwd=2, cex=0.8, bty='n')
```



```
bThr<-coords(rocTrn, "best", ret="threshold", transpose = FALSE)
bThr <- as.numeric(bThr)

bThr %>% view()
#Best threshold from ROC analyses
#bThr<-coords(rocTrn, "best", ret="threshold", transpose = FALSE)
#table(actual=revDTM_sentiBing_trn$hiLo, preds=revSentiBing_predTrn$predictions[,2]>0.5)
table(actual=revDTM_sentiBing_tst$hiLo, preds=revSentiBing_predTst$predictions[,2]>bThr)
```

```
##       preds
## actual FALSE  TRUE
##     -1  2672   872
##     1    906 10065
```

#SVM using Bing dictionary

```
library("e1071")
```

```
##
## Attaching package: 'e1071'
```

```
## The following object is masked from 'package:rsample':
##
##     permutations
```

```
library("ROCR")
#model 1
system.time( svmBing1 <- svm(as.factor(hiLo) ~., data = revDTM_sentiBing_trn
%>% select(-review_id), kernel="radial", cost=1, gamma=2, scale=FALSE, decision.values =
TRUE))
```

```
##    user  system elapsed
##  12.703   0.343  13.263
```

```
revDTM_predTrn_svmBing1<-predict(svmBing1, revDTM_sentiBing_trn, decision.values = TRUE)
table(actual= revDTM_sentiBing_trn$hiLo, predicted= revDTM_predTrn_svmBing1)
```

```
##       predicted
## actual    -1     1
##     -1  2423  1085
##     1    207 10801
```

```
revDTM_predTst_svmBing1<-predict(svmBing1, revDTM_sentiBing_tst, decision.values = TRUE)
table(actual= revDTM_sentiBing_tst$hiLo, predicted= revDTM_predTst_svmBing1)
```

```
##       predicted
## actual    -1     1
##     -1  2270  1274
##     1    335 10636
```

```
auc(as.numeric(revDTM_sentiBing_trn$hiLo), as.numeric(revDTM_predTrn_svmBing1))
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

```
## Area under the curve: 0.836
```

```
auc(as.numeric(revDTM_sentiBing_tst$hiLo), as.numeric(revDTM_predTst_svmBing1))
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.805
```

```
system.time( svmM2 <- svm(as.factor(hiLo) ~., data = revDTM_sentiBing_trn
%>% select(-review_id), kernel="radial", cost=5, gamma=5, scale=FALSE) )
```

```
##    user  system elapsed
##  19.694   0.444  20.892
```

```
revDTM_predTrn_svm2<-predict(svmM2, revDTM_sentiBing_trn)
table(actual= revDTM_sentiBing_trn$hiLo, predicted= revDTM_predTrn_svm2)
```

```
##       predicted
## actual    -1     1
##     -1  3053   455
##     1    107 10901
```

```
revDTM_predTst_svm2<-predict(svmM2, revDTM_sentiBing_tst)
table(actual= revDTM_sentiBing_tst$hiLo, predicted= revDTM_predTst_svm2)
```

```
##       predicted
## actual    -1     1
##     -1  2387  1157
##     1    507 10464
```

```
auc(as.numeric(revDTM_sentiBing_trn$hiLo), as.numeric(revDTM_predTrn_svm2))
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.9303
```

```
auc(as.numeric(revDTM_sentiBing_tst$hiLo), as.numeric(revDTM_predTst_svm2))
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.8137
```

#Naive Bayes with Bing Dictionary

```
library(pROC)
library(e1071)

#model 1
nbModel1<-naiveBayes(hiLo ~ ., data=revDTM_sentiBing_trn %>% select(-review_id))

#training data
revSentiBing_NBpredTrn<-predict(nbModel1, revDTM_sentiBing_trn, type = "raw")
cmtrn1 <- table(actual=revDTM_sentiBing_trn$hiLo, preds=revSentiBing_NBpredTrn[,2]>0.5)

auc(as.numeric(revDTM_sentiBing_trn$hiLo), revSentiBing_NBpredTrn[,2])
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

```
## Area under the curve: 0.6946
```

```
#nbModel1
#test data
revSentiBing_NBpredTst<-predict(nbModel1, revDTM_sentiBing_tst, type = "raw")
cmtst1 <- table(actual=revDTM_sentiBing_tst$hiLo, preds=revSentiBing_NBpredTst[,2]>0.5)

auc(as.numeric(revDTM_sentiBing_tst$hiLo), revSentiBing_NBpredTst[,2])
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.722
```

```
rocTrn <- roc(revDTM_sentiBing_trn$hiLo, revSentiBing_NBpredTrn[,2], levels=c(-1, 1))
```

```
## Setting direction: controls < cases
```

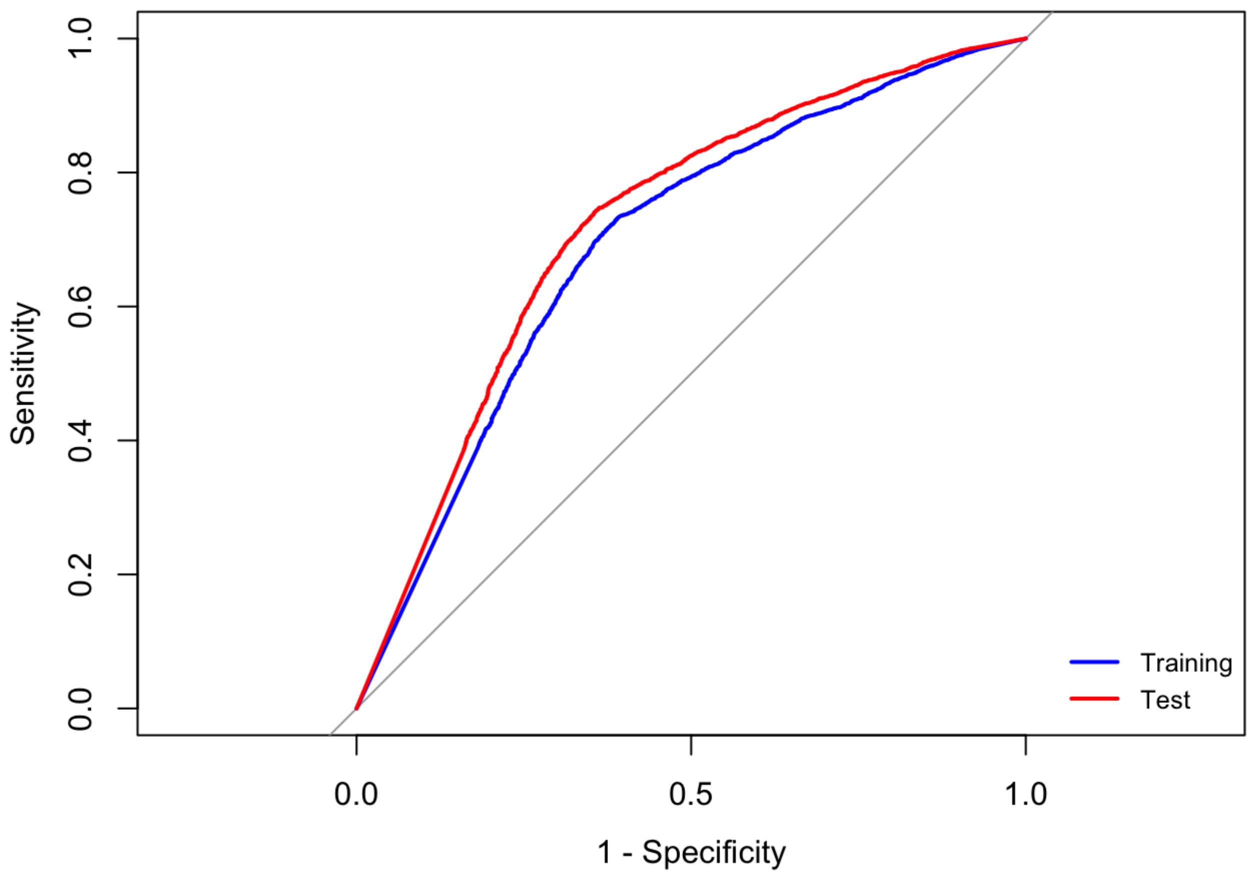
```
rocTst <- roc(revDTM_sentiBing_tst$hiLo, revSentiBing_NBpredTst[,2], levels=c(-1, 1))
```

```
## Setting direction: controls < cases
```

```
bThr<-coords(rocTrn, "best", ret="threshold", transpose = FALSE)
bThr <- as.numeric(bThr)

bThr %>% view()

table(actual=revDTM_sentiBing_tst$hiLo, preds=revSentiBing_NBpredTst[,2]>bThr)
```

```
##       preds
## actual FALSE TRUE
##     -1  2271 1273
##     1   2809 8162
```

```
plot.roc(rocTrn, col='blue', legacy.axes = TRUE)
plot.roc(rocTst, col='red', add=TRUE)
legend("bottomright", legend=c("Training", "Test"), col=c("blue", "red"), lwd=2, cex=0.8
, bty='n')
```



Sensitivity
1 - Specificity
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
Training
Test

##NRC###

```
#remove duplicates from rrSenti_nrc
rrSenti_nrc <-rrSenti_nrc[,-8]
rrSenti_nrc <-rrSenti_nrc[!duplicated(rrSenti_nrc), ]

#create Document Term Matrix
revDTM_sentiNrc <- rrSenti_nrc %>%  pivot_wider(id_cols = c(review_id,stars), names_from
= word, values_from = tf_idf)  %>% ungroup()


#filter out the reviews with stars=3
#calculate hiLo sentiment(1 is assigned to 4 and 5/-1 is assigned to 1 and 2)
revDTM_sentiNrc <- revDTM_sentiNrc %>% filter(stars!=3) %>% mutate(hiLo=ifelse(stars<=2,
-1, 1)) %>% select(-stars)

#replace all NAs with zero
revDTM_sentiNrc<-revDTM_sentiNrc %>% replace(., is.na(.), 0)

#convert hiLo from num to factor
revDTM_sentiNrc$hiLo<- as.factor(revDTM_sentiNrc$hiLo)


set.seed(1234)

#split the data into training and test dataset (50:50)
revDTM_sentiNrc_split<- initial_split(revDTM_sentiNrc, 0.5)
revDTM_sentiNrc_trn  <- training(revDTM_sentiNrc_split)
revDTM_sentiNrc_tst  <- testing(revDTM_sentiNrc_split)
```

##Ranger Model 1 with NRC

```
#RF Model
rfModel1<-ranger(dependent.variable.name = "hiLo", data=revDTM_sentiNrc_trn %>% select(-
review_id), num.trees = 500, importance='permutation', probability = TRUE)
```

```
## Growing trees.. Progress: 72%. Estimated remaining time: 12 seconds.
## Computing permutation importance.. Progress: 3%. Estimated remaining time: 15 minute
s, 37 seconds.
## Computing permutation importance.. Progress: 7%. Estimated remaining time: 16 minute
s, 58 seconds.
## Computing permutation importance.. Progress: 10%. Estimated remaining time: 15 minute
s, 48 seconds.
## Computing permutation importance.. Progress: 14%. Estimated remaining time: 14 minute
s, 24 seconds.
## Computing permutation importance.. Progress: 17%. Estimated remaining time: 13 minute
s, 47 seconds.
## Computing permutation importance.. Progress: 20%. Estimated remaining time: 13 minute
s, 12 seconds.
## Computing permutation importance.. Progress: 23%. Estimated remaining time: 12 minute
s, 32 seconds.
## Computing permutation importance.. Progress: 26%. Estimated remaining time: 12 minute
s, 10 seconds.
## Computing permutation importance.. Progress: 29%. Estimated remaining time: 11 minute
s, 46 seconds.
## Computing permutation importance.. Progress: 33%. Estimated remaining time: 11 minute
s, 5 seconds.
## Computing permutation importance.. Progress: 36%. Estimated remaining time: 10 minute
s, 23 seconds.
## Computing permutation importance.. Progress: 40%. Estimated remaining time: 9 minute
s, 46 seconds.
## Computing permutation importance.. Progress: 43%. Estimated remaining time: 9 minute
s, 16 seconds.
## Computing permutation importance.. Progress: 46%. Estimated remaining time: 8 minute
s, 40 seconds.
## Computing permutation importance.. Progress: 49%. Estimated remaining time: 8 minute
s, 13 seconds.
## Computing permutation importance.. Progress: 53%. Estimated remaining time: 7 minute
s, 42 seconds.
## Computing permutation importance.. Progress: 56%. Estimated remaining time: 7 minute
s, 14 seconds.
## Computing permutation importance.. Progress: 59%. Estimated remaining time: 6 minute
s, 44 seconds.
## Computing permutation importance.. Progress: 62%. Estimated remaining time: 6 minute
s, 13 seconds.
## Computing permutation importance.. Progress: 65%. Estimated remaining time: 5 minute
s, 41 seconds.
## Computing permutation importance.. Progress: 69%. Estimated remaining time: 5 minute
s, 7 seconds.
## Computing permutation importance.. Progress: 72%. Estimated remaining time: 4 minute
s, 33 seconds.
## Computing permutation importance.. Progress: 75%. Estimated remaining time: 4 minute
s, 3 seconds.
## Computing permutation importance.. Progress: 78%. Estimated remaining time: 3 minute
s, 31 seconds.
## Computing permutation importance.. Progress: 82%. Estimated remaining time: 3 minute
s, 0 seconds.
## Computing permutation importance.. Progress: 85%. Estimated remaining time: 2 minute
s, 29 seconds.
```

```
## Computing permutation importance.. Progress: 88%. Estimated remaining time: 1 minute,
57 seconds.
## Computing permutation importance.. Progress: 91%. Estimated remaining time: 1 minute,
26 seconds.
## Computing permutation importance.. Progress: 95%. Estimated remaining time: 52 second
s.
## Computing permutation importance.. Progress: 98%. Estimated remaining time: 21 second
s.
```

```
view(revDTM_sentiNrc_trn)

#Make predictions from the model on trn and test dataset
revSentiNrc_predTrn<- predict(rfModel1, revDTM_sentiNrc_trn %>% select(-review_id))
revSentiNrc_predTst<- predict(rfModel1, revDTM_sentiNrc_tst %>% select(-review_id))

bThr %>% view()
#best threshold from ROC
bThr<-coords(rocTrn, "best", ret="threshold", transpose = FALSE)
bThr <- as.numeric(bThr)

#Confusion Matrix at bThr for Trn and Tst dataset
a <- table(actual=revDTM_sentiNrc_trn$hiLo, preds=revSentiNrc_predTrn$predictions[,2]>bT
hr)
b <- table(actual=revDTM_sentiNrc_tst$hiLo, preds=revSentiNrc_predTst$predictions[,2]>bT
hr)

auc(as.numeric(revDTM_sentiNrc_trn$hiLo), revSentiNrc_predTrn$predictions[,2])
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

```
## Area under the curve: 0.9929
```

```
auc(as.numeric(revDTM_sentiNrc_tst$hiLo), revSentiNrc_predTst$predictions[,2])
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.9043
```

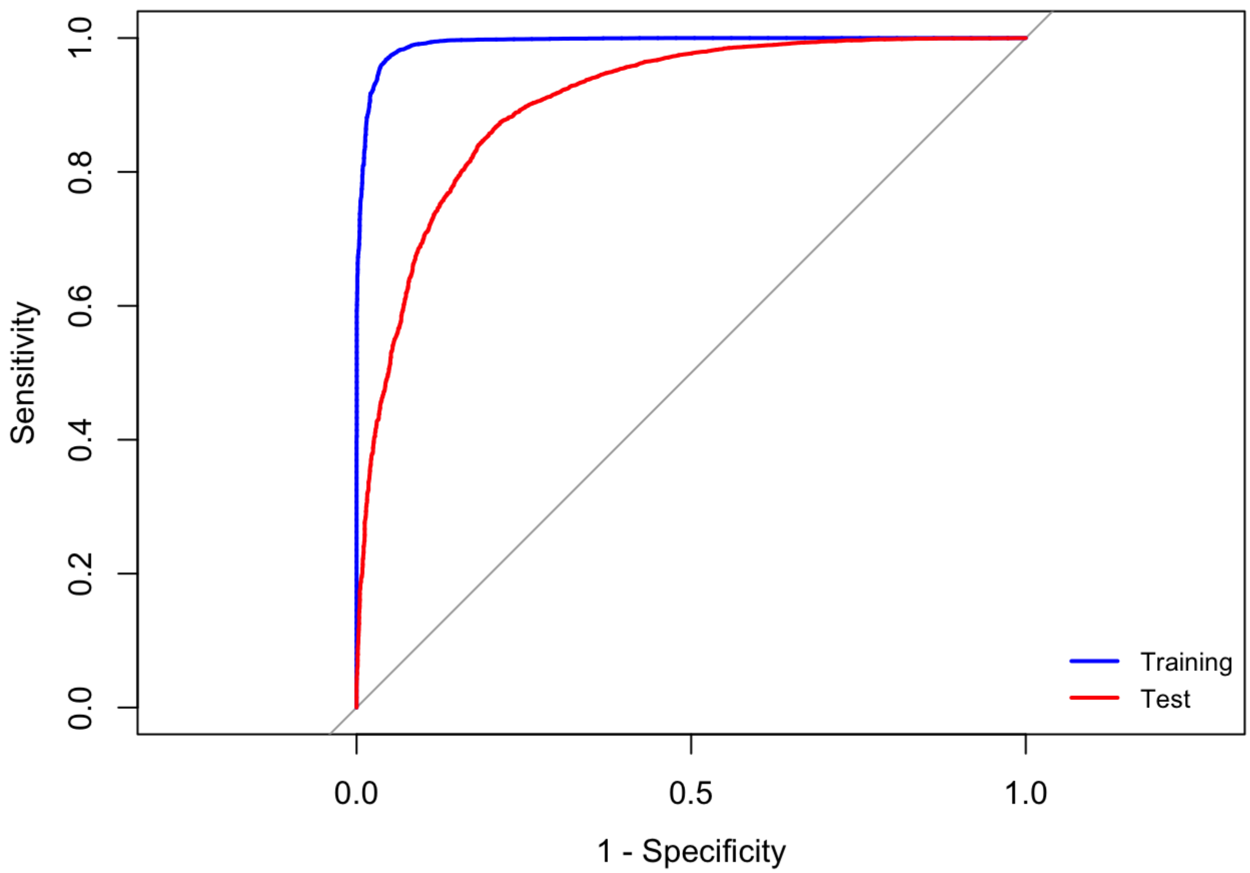
```
#importance(rfModel1) %>% view()

#rfModel1

library(pROC)
rocTrn <- roc(revDTM_sentiNrc_trn$hiLo, revSentiNrc_predTrn$predictions[,2], levels=c(-1
, 1))
```

```
## Setting direction: controls < cases
```

```
rocTst <- roc(revDTM_sentiNrc_tst$hiLo, revSentiNrc_predTst$predictions[,2], levels=c(-1
, 1))
```

```
## Setting direction: controls < cases
```

```
plot.roc(rocTrn, col='blue', legacy.axes = TRUE)
plot.roc(rocTst, col='red', add=TRUE)
legend("bottomright", legend=c("Training", "Test"),
        col=c("blue", "red"), lwd=2, cex=0.8, bty='n')
```



#SVM Models using NRC dictionary

```
#model 1
system.time( svmNRC1 <- svm(as.factor(hiLo) ~., data = revDTM_sentiNrc_trn
%>% select(-review_id), kernel="radial", cost=1, gamma=2, scale=FALSE, decision.values =
TRUE))
```

```
##    user  system elapsed
##  22.978   0.778  24.210
```

```
revDTM_predTrn_svmNRC1<-predict(svmNRC1, revDTM_sentiNrc_trn, decision.values = TRUE)
table(actual= revDTM_sentiNrc_trn$hiLo, predicted= revDTM_predTrn_svmNRC1)
```

```
##       predicted
## actual    -1     1
##     -1  2560  1076
##     1    199 11112
```

```
revDTM_predTst_svmNRC1<-predict(svmNRC1, revDTM_sentiNrc_tst, decision.values = TRUE)
table(actual= revDTM_sentiNrc_tst$hiLo, predicted= revDTM_predTst_svmNRC1)
```

```
##       predicted
## actual    -1     1
##     -1  2178  1520
##     1    321 10927
```

```
auc(as.numeric(revDTM_sentiNrc_trn$hiLo), as.numeric(revDTM_predTrn_svmNRC1))
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

```
## Area under the curve: 0.8432
```

```
auc(as.numeric(revDTM_sentiNrc_tst$hiLo), as.numeric(revDTM_predTst_svmNRC1))
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.7802
```

```
system.time( svmNRC2 <- svm(as.factor(hiLo) ~., data = revDTM_sentiNrc_trn
%>% select(-review_id), kernel="radial", cost=5, gamma=5, scale=FALSE, decision.values =
TRUE))
```

```
##    user  system elapsed
##  49.759   0.768  52.097
```

```
revDTM_predTrn_svmNRC2<-predict(svmNRC2, revDTM_sentiNrc_trn, decision.values = TRUE)
table(actual= revDTM_sentiNrc_trn$hiLo, predicted= revDTM_predTrn_svmNRC2)
```

```
##       predicted
## actual    -1     1
##     -1  3345   291
##     1     25 11286
```

```
revDTM_predTst_svmNRC2<-predict(svmNRC2, revDTM_sentiNrc_tst, decision.values = TRUE)
table(actual= revDTM_sentiNrc_tst$hiLo, predicted= revDTM_predTst_svmNRC2)
```

```
##       predicted
## actual    -1     1
##     -1  2135  1563
##     1    497 10751
```

```
auc(as.numeric(revDTM_sentiNrc_trn$hiLo), as.numeric(revDTM_predTrn_svmNRC2))
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.9589
```

```
auc(as.numeric(revDTM_sentiNrc_tst$hiLo), as.numeric(revDTM_predTst_svmNRC2))
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.7666
```

#Naive Bayes with NRC Dictionary

```
library(pROC)
library(e1071)

#model 1
nbModel1<-naiveBayes(hiLo ~ ., data=revDTM_sentiNrc_trn %>% select(-review_id))

#training data
revSentiNRC_NBpredTrn<-predict(nbModel1, revDTM_sentiNrc_trn, type = "raw")
cmtrn1 <- table(actual=revDTM_sentiNrc_trn$hiLo, preds=revSentiNRC_NBpredTrn[,2]>0.5)

auc(as.numeric(revDTM_sentiNrc_trn$hiLo), revSentiNRC_NBpredTrn[,2])
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

```
## Area under the curve: 0.673
```

```
#test data
revSentiNRC_NBpredTst<-predict(nbModel1, revDTM_sentiNrc_tst, type = "raw")
cmtst1 <- table(actual=revDTM_sentiNrc_tst$hiLo, preds=revSentiNRC_NBpredTst[,2]>0.5)

auc(as.numeric(revDTM_sentiNrc_tst$hiLo), revSentiNRC_NBpredTst[,2])
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.687
```

```
rocTrn <- roc(revDTM_sentiNrc_trn$hiLo, revSentiNRC_NBpredTrn[,2], levels=c(-1, 1))
```

```
## Setting direction: controls < cases
```

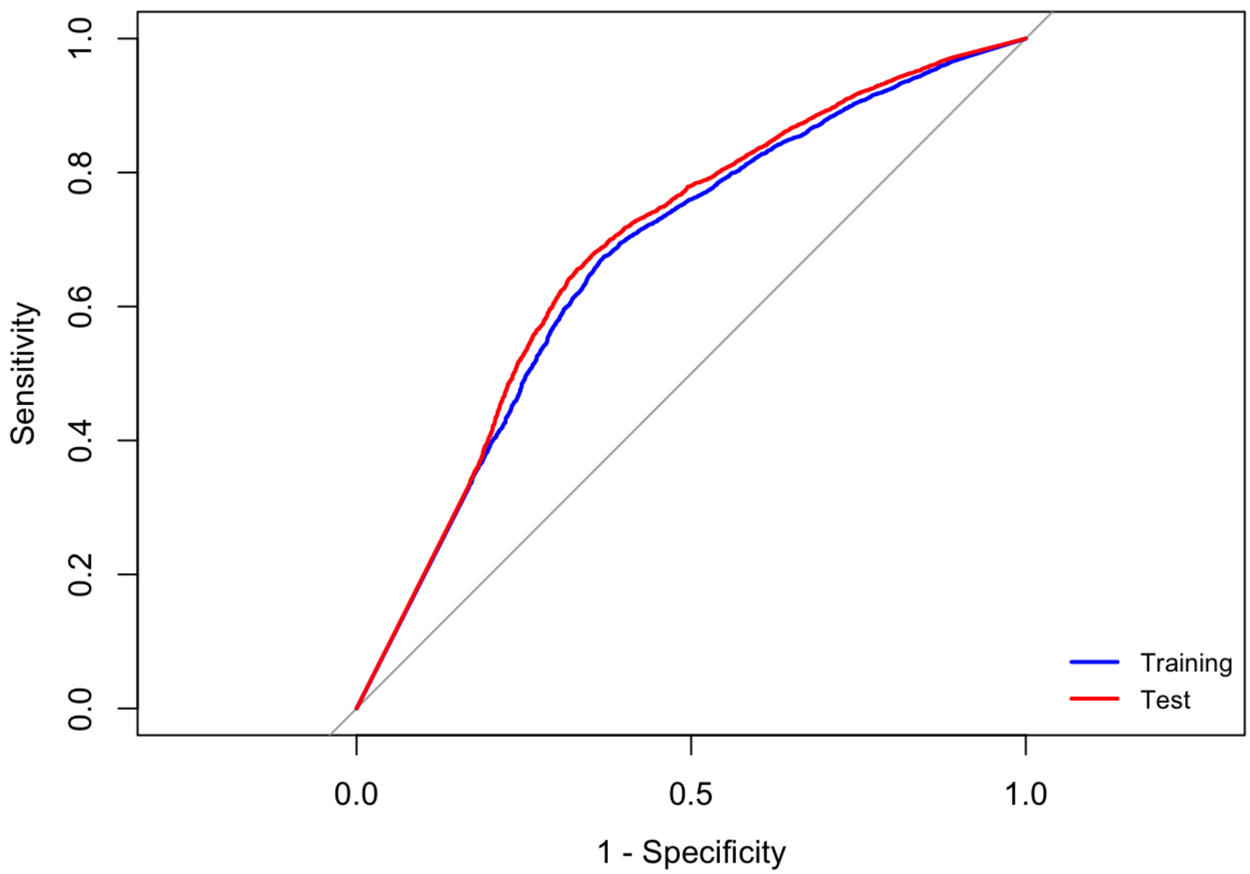
```
rocTst <- roc(revDTM_sentiNrc_tst$hiLo, revSentiNRC_NBpredTst[,2], levels=c(-1, 1))
```

```
## Setting direction: controls < cases
```

```
bThr<-coords(rocTrn, "best", ret="threshold", transpose = FALSE)
bThr <- as.numeric(bThr)

bThr %>% view()

table(actual=revDTM_sentiNrc_tst$hiLo, preds=revSentiNRC_NBpredTst[,2]>bThr)
```

```
##       preds
## actual FALSE TRUE
##     -1  2385 1313
##     1   3610 7638
```

```
plot.roc(rocTrn, col='blue', legacy.axes = TRUE)
plot.roc(rocTst, col='red', add=TRUE)
legend("bottomright", legend=c("Training", "Test"), col=c("blue", "red"), lwd=2, cex=0.8
, bty='n')
```



Afinn Dictionary

```
#create Document Term Matrix
revDTM_sentiAfinn <- rrSenti_afinn %>%  pivot_wider(id_cols = c(review_id,stars), names_
from = word, values_from = tf_idf)  %>% ungroup()

#filter out the reviews with stars=3
#calculate hiLo sentiment(1 is assigned to 4 and 5/-1 is assigned to 1 and 2)
revDTM_sentiAfinn <- revDTM_sentiAfinn %>% filter(stars!=3) %>% mutate(hiLo=ifelse(stars
<=2, -1, 1)) %>% select(-stars)

#replace all NAs with zero
revDTM_sentiAfinn<-revDTM_sentiAfinn %>% replace(., is.na(.), 0)

#convert hiLo from num to factor
revDTM_sentiAfinn$hiLo<- as.factor(revDTM_sentiAfinn$hiLo)


set.seed(1234)

#split the data into training and test dataset (50:50)
revDTM_sentiAfinn_split<- initial_split(revDTM_sentiAfinn, 0.5)
revDTM_sentiAfinn_trn  <- training(revDTM_sentiAfinn_split)
revDTM_sentiAfinn_tst  <- testing(revDTM_sentiAfinn_split)
```

#Random Forest models using Affin dictionary

```
#Model 1
rfModel1<-ranger(dependent.variable.name = "hiLo", data=revDTM_sentiAfinn_trn %>% select
(-review_id), num.trees = 300, importance='permutation', probability = TRUE)
```

```
## Computing permutation importance.. Progress: 21%. Estimated remaining time: 1 minute,
56 seconds.
## Computing permutation importance.. Progress: 42%. Estimated remaining time: 1 minute,
26 seconds.
## Computing permutation importance.. Progress: 62%. Estimated remaining time: 56 second
s.
## Computing permutation importance.. Progress: 83%. Estimated remaining time: 26 second
s.
```

```
#Make predictions from the model on trn and test dataset
revSentiAfinn_predTrn<- predict(rfModel1, revDTM_sentiAfinn_trn %>% select(-review_id))
revSentiAfinn_predTst<- predict(rfModel1, revDTM_sentiAfinn_tst %>% select(-review_id))

#find the optimal TH
rocTrn <- roc(revDTM_sentiAfinn_trn$hiLo, revSentiAfinn_predTrn$predictions[,2], levels=
c(-1, 1))
```

```
## Setting direction: controls < cases
```

```
rocTst <- roc(revDTM_sentiAfinn_tst$hiLo, revSentiAfinn_predTst$predictions[,2], levels=
c(-1, 1))
```

```
## Setting direction: controls < cases
```

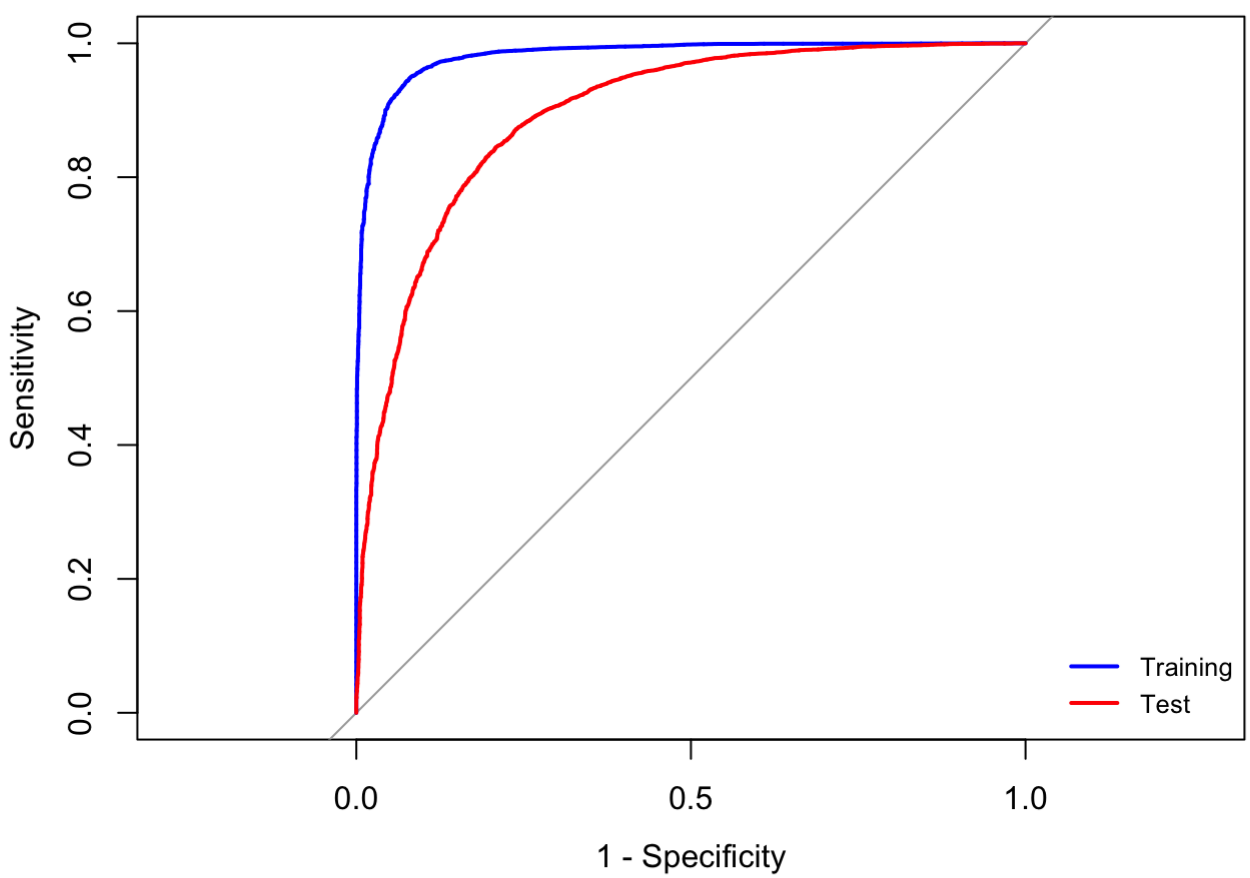
```
#best threshold from ROC
bThr<-coords(rocTrn, "best", ret="threshold", transpose = FALSE)
bThr <- as.numeric(bThr)

#Confusion Matrix at bThr for Trn and Tst dataset
a <- table(actual=revDTM_sentiAfinn_trn$hiLo, preds=revSentiAfinn_predTrn$predictions[,2
]>bThr)
b <- table(actual=revDTM_sentiAfinn_tst$hiLo, preds=revSentiAfinn_predTst$predictions[,2
]>bThr)

#a %>% view()

#find the optimal TH
rocTrn <- roc(revDTM_sentiAfinn_trn$hiLo, revSentiAfinn_predTrn$predictions[,2], levels=
c(-1, 1))
```

```
## Setting direction: controls < cases
```

```
rocTst <- roc(revDTM_sentiAfinn_tst$hiLo, revSentiAfinn_predTst$predictions[,2], levels=
c(-1, 1))
```

```
## Setting direction: controls < cases
```

```
#Best threshold from ROC analyses
#bThr<-coords(rocTrn, "best", ret="threshold", transpose = FALSE)
#table(actual=revDTM_sentiBing_trn$hiLo, preds=revSentiBing_predTrn[,2]>bThr)


library(pROC)
#rocTrn <- roc(revDTM_sentiBing_trn$hiLo, revSentiBing_predTrn[,2], levels=c(-1, 1))
#rocTst <- roc(revDTM_sentiBing_tst$hiLo, revSentiBing_predTst[,2], levels=c(-1, 1))

plot.roc(rocTrn, col='blue', legacy.axes = TRUE)
plot.roc(rocTst, col='red', add=TRUE)
legend("bottomright", legend=c("Training", "Test"),
        col=c("blue", "red"), lwd=2, cex=0.8, bty='n')
```




```
revSentiAfinn_predTrn %>% view()
#auc(as.numeric(revDTM_sentiAfinn_trn$hiLo), as.numeric(revSentiAfinn_predTrn))

#auc(as.numeric(revDTM_sentiAfinn_tst$hiLo), as.numeric(revSentiAfinn_predTst))
```

#Naive Bayes with AFINN Dictionary

```
library(pROC)
library(e1071)

#model 1
nbModel1<-naiveBayes(hiLo ~ ., data=revDTM_sentiAfinn_trn %>% select(-review_id))

#training data
revSentiAfinn_NBpredTrn<-predict(nbModel1, revDTM_sentiAfinn_trn, type = "raw")
cmtrn1 <- table(actual=revDTM_sentiAfinn_trn$hiLo, preds=revSentiAfinn_NBpredTrn[,2]>0.5
)

auc(as.numeric(revDTM_sentiAfinn_trn$hiLo), revSentiAfinn_NBpredTrn[,2])
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

```
## Area under the curve: 0.7284
```

```
#test data
revSentiAfinn_NBpredTst<-predict(nbModel1, revDTM_sentiAfinn_tst, type = "raw")
cmtst1 <- table(actual=revDTM_sentiAfinn_tst$hiLo, preds=revSentiAfinn_NBpredTst[,2]>0.5
)

auc(as.numeric(revDTM_sentiAfinn_tst$hiLo), revSentiAfinn_NBpredTst[,2])
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.7335
```

```
rocTrn <- roc(revDTM_sentiAfinn_trn$hiLo, revSentiAfinn_NBpredTrn[,2], levels=c(-1, 1))
```

```
## Setting direction: controls < cases
```

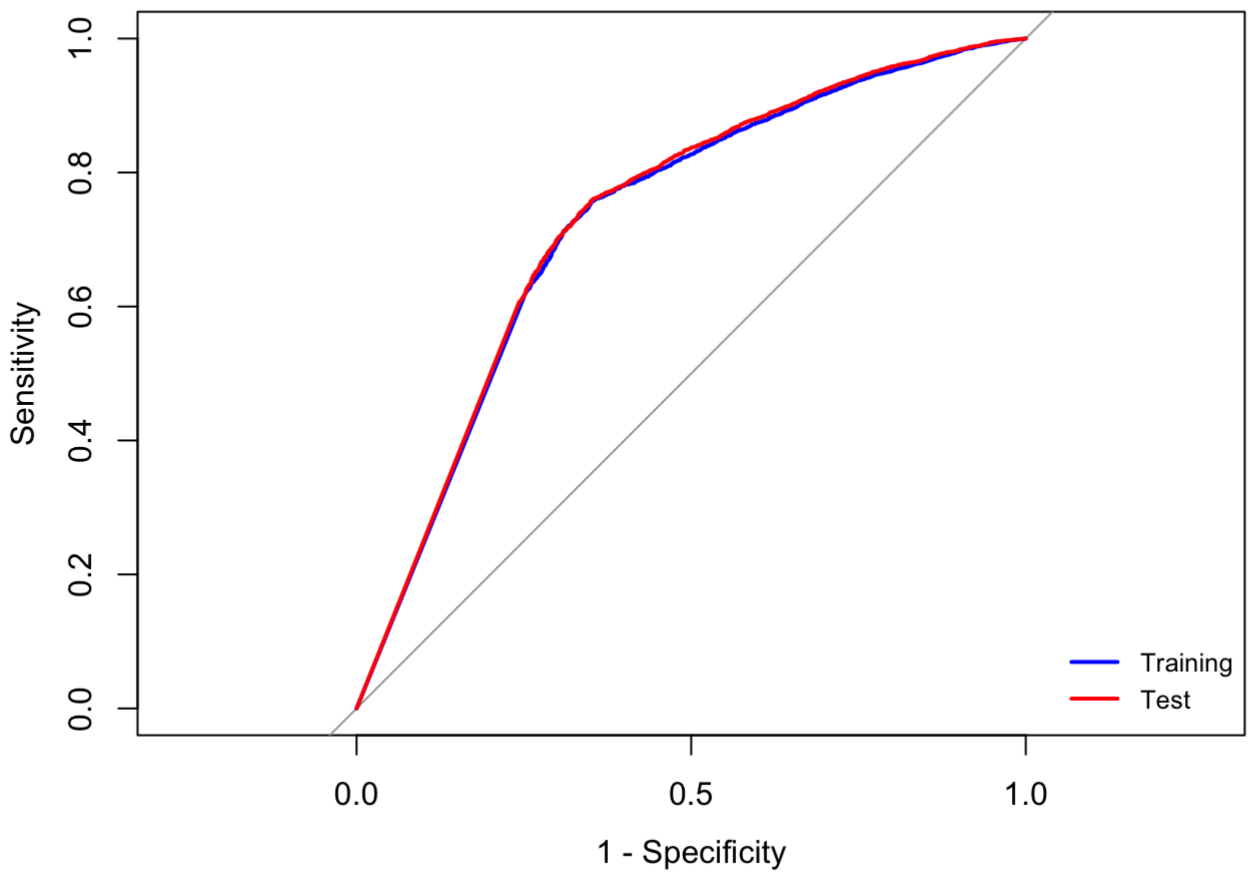
```
rocTst <- roc(revDTM_sentiAfinn_tst$hiLo, revSentiAfinn_NBpredTst[,2], levels=c(-1, 1))
```

```
## Setting direction: controls < cases
```

```
bThr<-coords(rocTrn, "best", ret="threshold", transpose = FALSE)
bThr <- as.numeric(bThr)

bThr %>% view()

table(actual=revDTM_sentiAfinn_tst$hiLo, preds=revSentiAfinn_NBpredTst[,2]>bThr)
```

```
##       preds
## actual FALSE TRUE
##     -1  2235 1216
##     1   2598 8176
```

```
plot.roc(rocTrn, col='blue', legacy.axes = TRUE)
plot.roc(rocTst, col='red', add=TRUE)
legend("bottomright", legend=c("Training", "Test"), col=c("blue", "red"), lwd=2, cex=0.8
, bty='n')
```



#SVM Model

```
#model 1
system.time(svmAfinn1 <- svm(as.factor(hiLo) ~., data = revDTM_sentiAfinn_trn
%>% select(-review_id), kernel="radial", cost=1, gamma = 1,scale=FALSE, decision.values
 = TRUE))
```

```
##    user  system elapsed
##   9.008   0.432   9.991
```

```
revDTM_predTrn_svmAfinn1<-predict(svmAfinn1, revDTM_sentiAfinn_trn, decision.values = TR
UE)
table(actual= revDTM_sentiAfinn_trn$hiLo, predicted= revDTM_predTrn_svmAfinn1)
```

```
##       predicted
## actual    -1     1
##     -1  1912  1548
##     1    273 10492
```

```
revDTM_predTst_svmAfinn1<-predict(svmAfinn1, revDTM_sentiAfinn_tst, decision.values = TR
UE)
table(actual= revDTM_sentiAfinn_tst$hiLo, predicted= revDTM_predTst_svmAfinn1)
```

```
##       predicted
## actual    -1     1
##     -1  1778  1673
##     1    310 10464
```

```
auc(as.numeric(revDTM_sentiAfinn_trn$hiLo), as.numeric(revDTM_predTrn_svmAfinn1))
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

```
## Area under the curve: 0.7636
```

```
auc(as.numeric(revDTM_sentiAfinn_tst$hiLo), as.numeric(revDTM_predTst_svmAfinn1))
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.7432
```

```
#model 2
system.time(svmAfinn2 <- svm(as.factor(hiLo) ~., data = revDTM_sentiAfinn_trn
%>% select(-review_id), kernel="radial", cost=5, gamma = 5, scale=FALSE, decision.values
= TRUE))
```

```
##    user  system elapsed
##  13.065   0.289  13.702
```

```
revDTM_predTrn_svmAfinn2<-predict(svmAfinn2, revDTM_sentiAfinn_trn, decision.values = TR
UE)
table(actual= revDTM_sentiAfinn_trn$hiLo, predicted= revDTM_predTrn_svmAfinn2)
```

```
##       predicted
## actual    -1     1
##     -1  2679   781
##     1    209 10556
```

```
revDTM_predTst_svmAfinn2<-predict(svmAfinn2, revDTM_sentiAfinn_tst, decision.values = TR
UE)
table(actual= revDTM_sentiAfinn_tst$hiLo, predicted= revDTM_predTst_svmAfinn2)
```

```
##       predicted
## actual    -1     1
##     -1  2094  1357
##     1    534 10240
```

```
auc(as.numeric(revDTM_sentiAfinn_trn$hiLo), as.numeric(revDTM_predTrn_svmAfinn2))
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.8774
```

```
auc(as.numeric(revDTM_sentiAfinn_tst$hiLo), as.numeric(revDTM_predTst_svmAfinn2))
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.7786
```

####Broader set of Terms Models ####

```
#if we want to remove the words which are there in too many or too few of the reviews
#First find out how many reviews each word occurs in
rWords<-rrTokens %>% group_by(word) %>% summarise(nr=n()) %>% arrange(desc(nr))

#How many words are there
length(rWords$word)
```

```
## [1] 7175
```

```
top_n(rWords, 20)
```

word <chr>	**nr** <int>
food	17089
service	10782
time	8916
eat	6534
restaurant	6479
love	5922
price	5188
nice	5059
chicken	5048
delicious	4836

1-10 of 20 rows Previous **1** 2 Next

```
top_n(rWords, -20)
```

word <chr>	**nr** <int>
angkor	6
chao	6
chin	6
elia	6
kashmir	6
kielbasa	6
paymons	6
rt	6
rudys	6
shimogamo	6

1-10 of 36 rows Previous **1** 2 3 4 Next

```
#Suppose we want to remove words which occur in > 90% of reviews, and those which are i
n, for example, less than 30 reviews
reduced_rWords<-rWords %>% filter(nr< 6000 & nr > 30)
length(reduced_rWords$word)
```

```
## [1] 3415
```

```
#reduce the rrTokens data to keep only the reduced set of words
reduced_rrTokens <- left_join(reduced_rWords, rrTokens)

#Now convert it to a DTM, where each row is for a review (document), and columns are the
terms (words)
revDTM  <- reduced_rrTokens %>%  pivot_wider(id_cols = c(review_id,stars), names_from =
 word, values_from = tf_idf)  %>% ungroup()

#Check
dim(revDTM)
```

```
## [1] 35310  3417
```

```
  #do the numberof columsnmatch the words -- we should also have the stars column and th
e review_id

#create the dependent variable hiLo of good/bad reviews absed on stars, and remove the r
eview with stars=3
revDTM <- revDTM %>% filter(stars!=3) %>% mutate(hiLo=ifelse(stars<=2, -1, 1)) %>% selec
t(-stars)

#replace NAs with 0s
revDTM<-revDTM %>% replace(., is.na(.), 0)

revDTM$hiLo<-as.factor(revDTM$hiLo)

revDTM_split<- initial_split(revDTM, 0.5)
revDTM_trn<- training(revDTM_split)
revDTM_tst<- testing(revDTM_split)

#this can take some time...the importance ='permutation' takes time (we know why)
rfModel2<-ranger(dependent.variable.name = "hiLo", data=revDTM_trn %>% select(-review_i
d), num.trees = 200, importance='permutation', probability = TRUE)
```

```
## Computing permutation importance.. Progress: 1%. Estimated remaining time: 1 hour, 52
minutes, 46 seconds.
## Computing permutation importance.. Progress: 5%. Estimated remaining time: 26 minute
s, 52 seconds.
## Computing permutation importance.. Progress: 9%. Estimated remaining time: 20 minute
s, 37 seconds.
## Computing permutation importance.. Progress: 13%. Estimated remaining time: 17 minute
s, 58 seconds.
## Computing permutation importance.. Progress: 17%. Estimated remaining time: 16 minute
s, 16 seconds.
## Computing permutation importance.. Progress: 21%. Estimated remaining time: 14 minute
s, 36 seconds.
## Computing permutation importance.. Progress: 25%. Estimated remaining time: 13 minute
s, 27 seconds.
## Computing permutation importance.. Progress: 28%. Estimated remaining time: 12 minute
s, 57 seconds.
## Computing permutation importance.. Progress: 30%. Estimated remaining time: 12 minute
s, 45 seconds.
## Computing permutation importance.. Progress: 34%. Estimated remaining time: 12 minute
s, 2 seconds.
## Computing permutation importance.. Progress: 38%. Estimated remaining time: 11 minute
s, 13 seconds.
## Computing permutation importance.. Progress: 42%. Estimated remaining time: 10 minute
s, 31 seconds.
## Computing permutation importance.. Progress: 46%. Estimated remaining time: 9 minute
s, 48 seconds.
## Computing permutation importance.. Progress: 50%. Estimated remaining time: 8 minute
s, 59 seconds.
## Computing permutation importance.. Progress: 53%. Estimated remaining time: 8 minute
s, 27 seconds.
## Computing permutation importance.. Progress: 55%. Estimated remaining time: 8 minute
s, 15 seconds.
## Computing permutation importance.. Progress: 58%. Estimated remaining time: 7 minute
s, 33 seconds.
## Computing permutation importance.. Progress: 62%. Estimated remaining time: 6 minute
s, 55 seconds.
## Computing permutation importance.. Progress: 66%. Estimated remaining time: 6 minute
s, 11 seconds.
## Computing permutation importance.. Progress: 70%. Estimated remaining time: 5 minute
s, 28 seconds.
## Computing permutation importance.. Progress: 73%. Estimated remaining time: 4 minute
s, 48 seconds.
## Computing permutation importance.. Progress: 77%. Estimated remaining time: 4 minute
s, 10 seconds.
## Computing permutation importance.. Progress: 80%. Estimated remaining time: 3 minute
s, 38 seconds.
## Computing permutation importance.. Progress: 82%. Estimated remaining time: 3 minute
s, 13 seconds.
## Computing permutation importance.. Progress: 86%. Estimated remaining time: 2 minute
s, 36 seconds.
## Computing permutation importance.. Progress: 89%. Estimated remaining time: 1 minute,
57 seconds.
## Computing permutation importance.. Progress: 92%. Estimated remaining time: 1 minute,
```

```
26 seconds.
## Computing permutation importance.. Progress: 96%. Estimated remaining time: 42 second
s.
## Computing permutation importance.. Progress: 100%. Estimated remaining time: 5 second
s.
```

```
rfModel2
```

```
## Ranger result
##
## Call:
##  ranger(dependent.variable.name = "hiLo", data = revDTM_trn %>%      select(-review_i
d), num.trees = 200, importance = "permutation",      probability = TRUE)
##
## Type:                             Probability estimation
## Number of trees:                  200
## Sample size:                      15166
## Number of independent variables:  3415
## Mtry:                             58
## Target node size:                 10
## Variable importance mode:         permutation
## Splitrule:                        gini
## OOB prediction error (Brier s.):  0.08656238
```

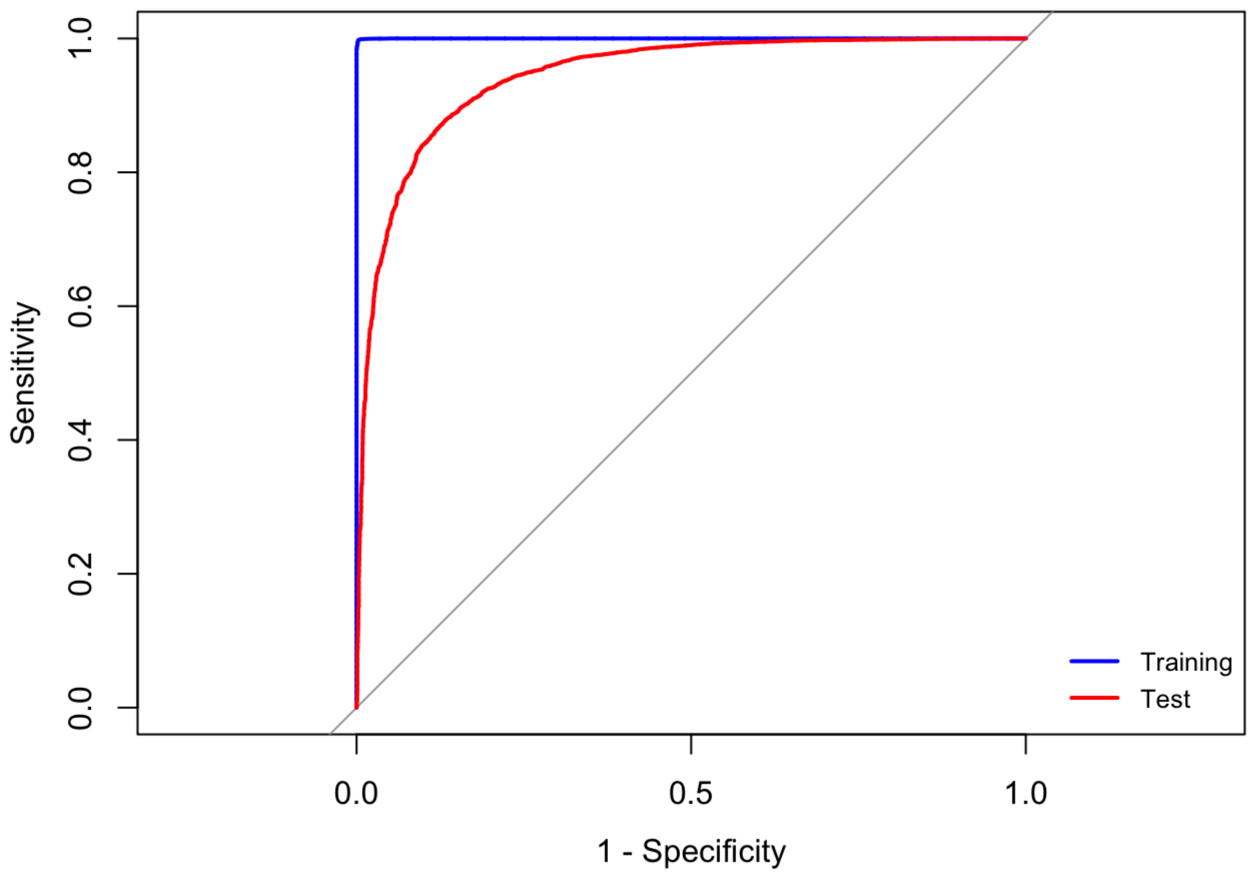
```
revSentiNDict_predTrn<- predict(rfModel2, revDTM_trn %>% select(-review_id))
revSentiNDict_predTst<- predict(rfModel2, revDTM_tst %>% select(-review_id))


importance(rfModel2) %>% view()

#rfModel1

library(pROC)
rocTrn <- roc(revDTM_trn$hiLo, revSentiNDict_predTrn$predictions[,2], levels=c(-1, 1))
rocTst <- roc(revDTM_tst$hiLo, revSentiNDict_predTst$predictions[,2], levels=c(-1, 1))

#bThr %>% view()
#best threshold from ROC
bThr<-coords(rocTrn, "best", ret="threshold", transpose = FALSE)
bThr <- as.numeric(bThr)
bThr %>% view()


#Confusion Matrix at bThr for Trn and Tst dataset
a <- table(actual=revDTM_trn$hiLo, preds=revSentiNDict_predTrn$predictions[,2]>bThr)
b <- table(actual=revDTM_tst$hiLo, preds=revSentiNDict_predTst$predictions[,2]>bThr)

auc(as.numeric(revDTM_trn$hiLo), revSentiNDict_predTrn$predictions[,2])
```

```
## Area under the curve: 1
```

```
auc(as.numeric(revDTM_tst$hiLo), revSentiNDict_predTst$predictions[,2])
```

```
## Area under the curve: 0.9445
```

```
plot.roc(rocTrn, col='blue', legacy.axes = TRUE)
plot.roc(rocTst, col='red', add=TRUE)
legend("bottomright", legend=c("Training", "Test"),
        col=c("blue", "red"), lwd=2, cex=0.8, bty='n')
```



Naive Bayes On broder terms

```
library(pROC)
library(e1071)

#model 1
nbModel1<-naiveBayes(hiLo ~ ., data=revDTM_trn %>% select(-review_id))

#training data
revSentiNdict_NBpredTrn<-predict(nbModel1, revDTM_trn, type = "raw")
cmtrn1 <- table(actual=revDTM_trn$hiLo, preds=revSentiNdict_NBpredTrn[,2]>0.5)

auc(as.numeric(revDTM_trn$hiLo), revSentiNdict_NBpredTrn[,2])
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

```
## Area under the curve: 0.69
```

```
#test data
revSentiNdict_NBpredTst<-predict(nbModel1, revDTM_tst, type = "raw")
cmtst1 <- table(actual=revDTM_tst$hiLo, preds=revSentiNdict_NBpredTst[,2]>0.5)

auc(as.numeric(revDTM_tst$hiLo), revSentiNdict_NBpredTst[,2])
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.7152
```

```
rocTrn <- roc(revDTM_trn$hiLo, revSentiNdict_NBpredTrn[,2], levels=c(-1, 1))
```

```
## Setting direction: controls < cases
```

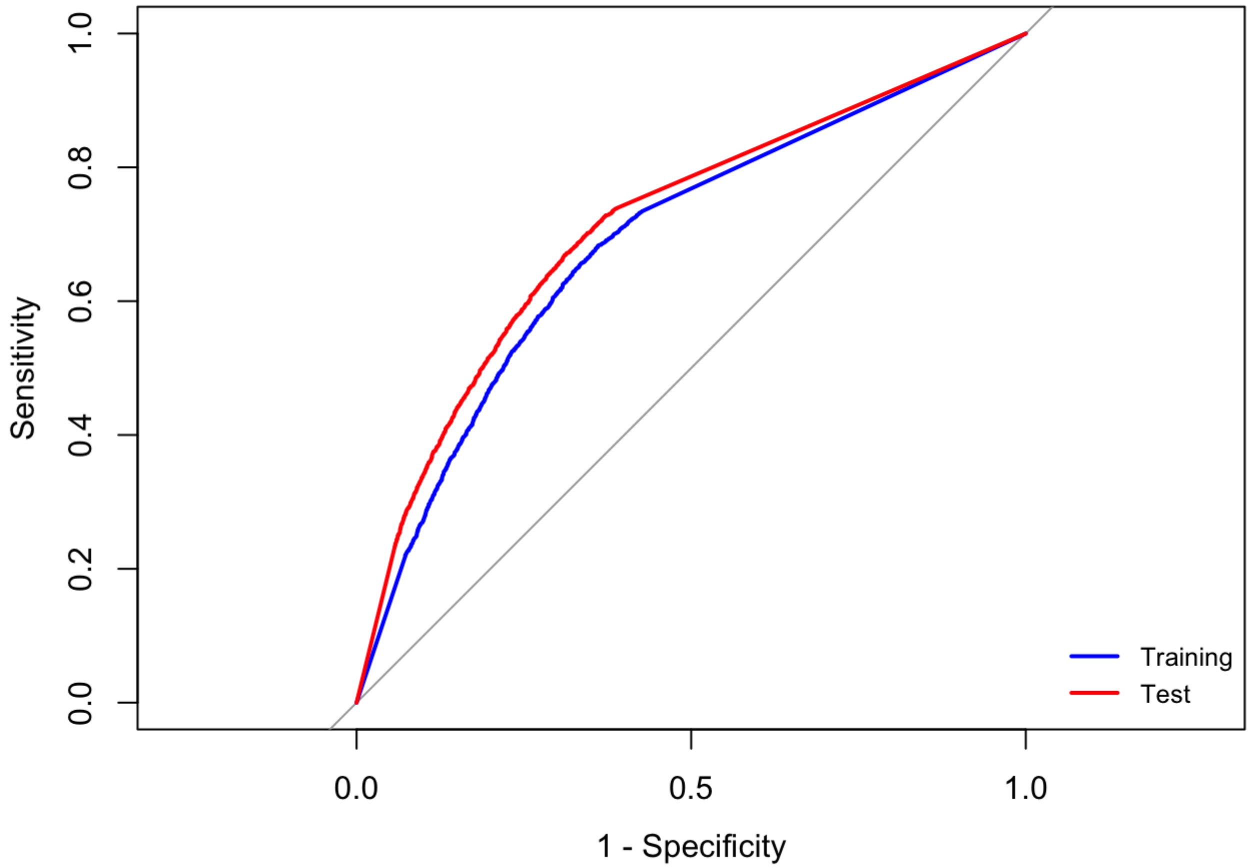
```
rocTst <- roc(revDTM_tst$hiLo, revSentiNdict_NBpredTst[,2], levels=c(-1, 1))
```

```
## Setting direction: controls < cases
```

```
bThr<-coords(rocTrn, "best", ret="threshold", transpose = FALSE)
bThr <- as.numeric(bThr)

bThr %>% view()

table(actual=revDTM_tst$hiLo, preds=revSentiNdict_NBpredTst[,2]>bThr)
```

```
##       preds
## actual FALSE TRUE
##     -1  2469 1228
##     1   3577 7892
```

```
plot.roc(rocTrn, col='blue', legacy.axes = TRUE)
plot.roc(rocTst, col='red', add=TRUE)
legend("bottomright", legend=c("Training", "Test"), col=c("blue", "red"), lwd=2, cex=0.8
, bty='n')
```



#SVM Model

```
#model 1
system.time(svmNDict1 <- svm(as.factor(hiLo) ~., data = revDTM_trn
%>% select(-review_id), kernel="radial", cost=1, gamma = 1,scale=FALSE, decision.values
 = TRUE))
```

```
##    user  system elapsed 
##  76.665   4.414  92.226
```

```
revDTM_predTrn_svmNDict1<-predict(svmNDict1, revDTM_trn, decision.values = TRUE)
table(actual= revDTM_trn$hiLo, predicted= revDTM_predTrn_svmNDict1)
```

```
##       predicted
## actual    -1     1
##     -1  3385   340
##     1     45 11396
```

```
revDTM_predTst_svmNDict1<-predict(svmNDict1, revDTM_tst, decision.values = TRUE)
table(actual= revDTM_tst$hiLo, predicted= revDTM_predTst_svmNDict1)
```

```
##       predicted
## actual    -1     1
##     -1  2515  1182
##     1    242 11227
```

```
auc(as.numeric(revDTM_trn$hiLo), as.numeric(revDTM_predTrn_svmNDict1))
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

```
## Area under the curve: 0.9524
```

```
auc(as.numeric(revDTM_tst$hiLo), as.numeric(revDTM_predTst_svmNDict1))
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.8296
```

```
#model 2
system.time(svmNDict2 <- svm(as.factor(hiLo) ~., data = revDTM_trn
%>% select(-review_id), kernel="radial", cost=5, gamma = 5,scale=FALSE, decision.values
 = TRUE))
```

```
##    user  system elapsed
## 170.040   5.482 190.017
```

```
revDTM_predTrn_svmNDict2<-predict(svmNDict2, revDTM_trn, decision.values = TRUE)
table(actual= revDTM_trn$hiLo, predicted= revDTM_predTrn_svmNDict2)
```

```
##       predicted
## actual    -1     1
##     -1  3725     0
##     1      0 11441
```

```
revDTM_predTst_svmNDict2<-predict(svmNDict2, revDTM_tst, decision.values = TRUE)
table(actual= revDTM_tst$hiLo, predicted= revDTM_predTst_svmNDict2)
```

```
##       predicted
## actual    -1     1
##     -1   847  2850
##     1     62 11407
```

```
auc(as.numeric(revDTM_trn$hiLo), as.numeric(revDTM_predTrn_svmNDict2))
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 1
```

```
auc(as.numeric(revDTM_tst$hiLo), as.numeric(revDTM_predTst_svmNDict2))
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.6118
```

```
# 4) Combined dictionary
# Preparing the Document Term Matrix

#combine; create rrTokens_com
rrTokens_com <- rrTokens %>% left_join(get_sentiments("bing"), by="word")
colnames(rrTokens_com)[8] <- "senti.bing"
rrTokens_com <- rrTokens_com %>% left_join(get_sentiments("nrc"), by="word")
colnames(rrTokens_com)[9] <- "senti.nrc"
rrTokens_com <- rrTokens_com %>% left_join(get_sentiments("afinn"), by="word")
colnames(rrTokens_com)[10] <- "senti.afinn"

#mutate hiLo
rrTokens_com <- rrTokens_com %>% mutate(hiLo=ifelse(stars<=2,-1, ifelse(stars>=4, 1, 0
 )))
#mutate hiLo.bing
rrTokens_com <- rrTokens_com %>% mutate(hiLo.bing=ifelse(senti.bing=="positive", 1, -1))
#mutate hiLo.nrc
rrTokens_com <- rrTokens_com  %>% mutate(hiLo.nrc=ifelse(senti.nrc %in% c('anger', 'disg
ust', 'fear', 'sadness', 'negative'), -1,ifelse(senti.nrc %in% c('positive', 'joy', 'ant
icipation', 'trust'), 1, 0)))
#mutate hiLo.afinn
rrTokens_com <- rrTokens_com %>% mutate(hiLo.afinn=ifelse(senti.afinn >0, 1, -1))
rrTokens_com <- rrTokens_com %>% select(-senti.bing, -senti.nrc,-senti.afinn)

#replace NA with 0
rrTokens_com <- rrTokens_com %>% replace(., is.na(.), 0)
#combine 3 dictionaries
rrTokens_com <- rrTokens_com %>% mutate(hiLo.com = hiLo.bing+hiLo.nrc+hiLo.afinn)
#mutate comm
rrTokens_com <- rrTokens_com %>% mutate(hiLo.comm=ifelse(hiLo.com>0,1,ifelse(hiLo.com<0,
-1, 0 )))
#filter out unmatch words
rrTokens_com <- rrTokens_com %>% filter(hiLo.comm != 0)

#for pivot
m <- rrTokens_com %>% select(-n,-tf,-idf,-hiLo.bing,-hiLo.nrc,-hiLo.afinn,,-hiLo.com,-hi
Lo,-hiLo.com,-hiLo.comm) %>% distinct()
dim(m)
```

```
## [1] 357370      4
```

```
#pivot table
revDTM_com <- m %>%pivot_wider(id_cols = c(review_id,stars), names_from = word, values_f
rom = tf_idf) %>% ungroup()
dim(revDTM_com)
```

```
## [1] 35016  1919
```

```
#filter out the reviews with stars=3, and calculate hiLo sentiment 'class'
revDTM_com <- revDTM_com %>% filter(stars!=3) %>% mutate(hiLo=ifelse(stars<=2, -1, 1)) %
>% select(-stars)

#replace all the NAs with 0
revDTM_com <- revDTM_com %>% replace(., is.na(.), 0)

#change to factor
revDTM_com$hiLo <- as.factor(revDTM_com$hiLo)

library(dplyr)
#class(rrSenti_bing)
set.seed(1789)
dim(revDTM_com)
```

```
## [1] 30066  1919
```

```
revDTM_com_10k <- revDTM_com[sample(nrow(revDTM_com), 10000), ]

library(rsample)
set.seed(1789)
revDTM_com_split<- initial_split(revDTM_com_10k, 0.5)
revDTM_com_trn<- training(revDTM_com_split)
revDTM_com_tst<- testing(revDTM_com_split)

########### Compare to stars ###############
xx<- rrTokens_com %>% filter(hiLo!=0)
table(actual=xx$hiLo, predicted=xx$hiLo.comm)
```

```
##       predicted
## actual     -1      1
##     -1  69819  86877
##     1   99871 341603
```

```
#Random Forest 1

rfModel1<-ranger(dependent.variable.name = "hiLo", data=revDTM_com_trn %>% select(-revie
w_id), num.trees = 200, importance='permutation', probability = TRUE)
```

```
## Computing permutation importance.. Progress: 33%. Estimated remaining time: 1 minute,
4 seconds.
## Computing permutation importance.. Progress: 65%. Estimated remaining time: 33 second
s.
## Computing permutation importance.. Progress: 97%. Estimated remaining time: 2 second
s.
```

```
#Make predictions from the model on trn and test dataset
combined_dict_DTM_predTrn<- predict(rfModel1, revDTM_com_trn %>% select(-review_id))
combined_dict_DTM_predTst<- predict(rfModel1, revDTM_com_tst %>% select(-review_id))

bThr %>% view()
#best threshold from ROC
bThr<-coords(rocTrn, "best", ret="threshold", transpose = FALSE)
bThr <- as.numeric(bThr)

#Confusion Matrix at bThr for Trn and Tst dataset
a <- table(actual=revDTM_com_trn$hiLo, preds=combined_dict_DTM_predTrn$predictions[,2]>b
Thr)
b <- table(actual=revDTM_com_tst$hiLo, preds=combined_dict_DTM_predTst$predictions[,2]>b
Thr)

auc(as.numeric(revDTM_com_trn$hiLo), combined_dict_DTM_predTrn$predictions[,2])
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

```
## Area under the curve: 0.9967
```

```
auc(as.numeric(revDTM_com_tst$hiLo), combined_dict_DTM_predTst$predictions[,2])
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.9094
```

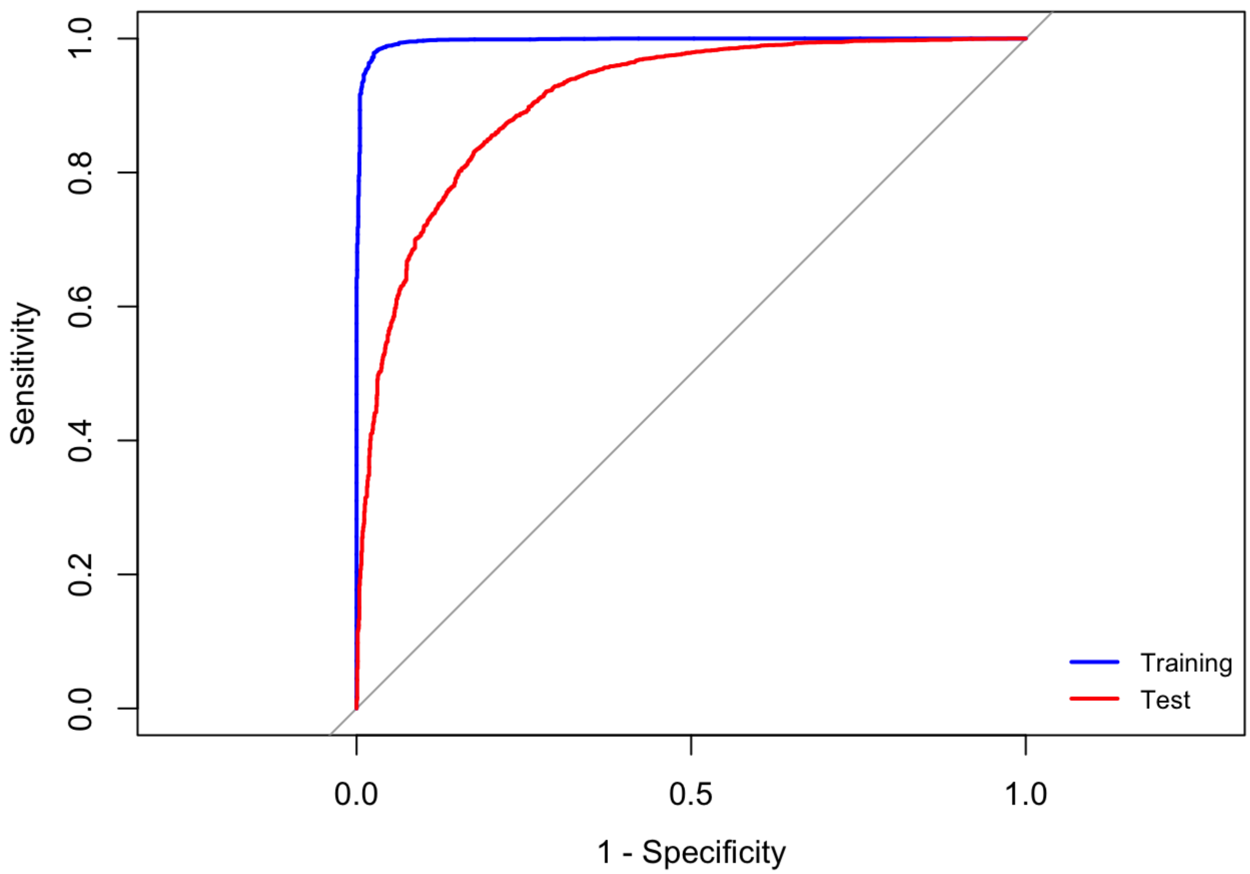
```
#importance(rfModel1) %>% view()

#rfModel1

library(pROC)
rocTrn <- roc(revDTM_com_trn$hiLo, combined_dict_DTM_predTrn$predictions[,2], levels=c(-
1, 1))
```

```
## Setting direction: controls < cases
```

```
rocTst <- roc(revDTM_com_tst$hiLo, combined_dict_DTM_predTst$predictions[,2], levels=c(-
1, 1))
```

```
## Setting direction: controls < cases
```

```
plot.roc(rocTrn, col='blue', legacy.axes = TRUE)
plot.roc(rocTst, col='red', add=TRUE)
legend("bottomright", legend=c("Training", "Test"),
        col=c("blue", "red"), lwd=2, cex=0.8, bty='n')
```




```
#Naive Bayes on Combined dictionary

library(pROC)
library(e1071)

#model 1
nbModel1<-naiveBayes(hiLo ~ ., data=revDTM_com_trn %>% select(-review_id))

#training data
revSentiNdict_NBpredTrn<-predict(nbModel1, revDTM_com_trn, type = "raw")
cmtrn1 <- table(actual=revDTM_com_trn$hiLo, preds=revSentiNdict_NBpredTrn[,2]>0.5)

auc(as.numeric(revDTM_com_trn$hiLo), revSentiNdict_NBpredTrn[,2])
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

```
## Area under the curve: 0.5095
```

```
#test data
revSentiNdict_NBpredTst<-predict(nbModel1, revDTM_com_tst, type = "raw")
cmtst1 <- table(actual=revDTM_com_tst$hiLo, preds=revSentiNdict_NBpredTst[,2]>0.5)

auc(as.numeric(revDTM_com_tst$hiLo), revSentiNdict_NBpredTst[,2])
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.6804
```

```
rocTrn <- roc(revDTM_com_trn$hiLo, revSentiNdict_NBpredTrn[,2], levels=c(-1, 1))
```

```
## Setting direction: controls < cases
```

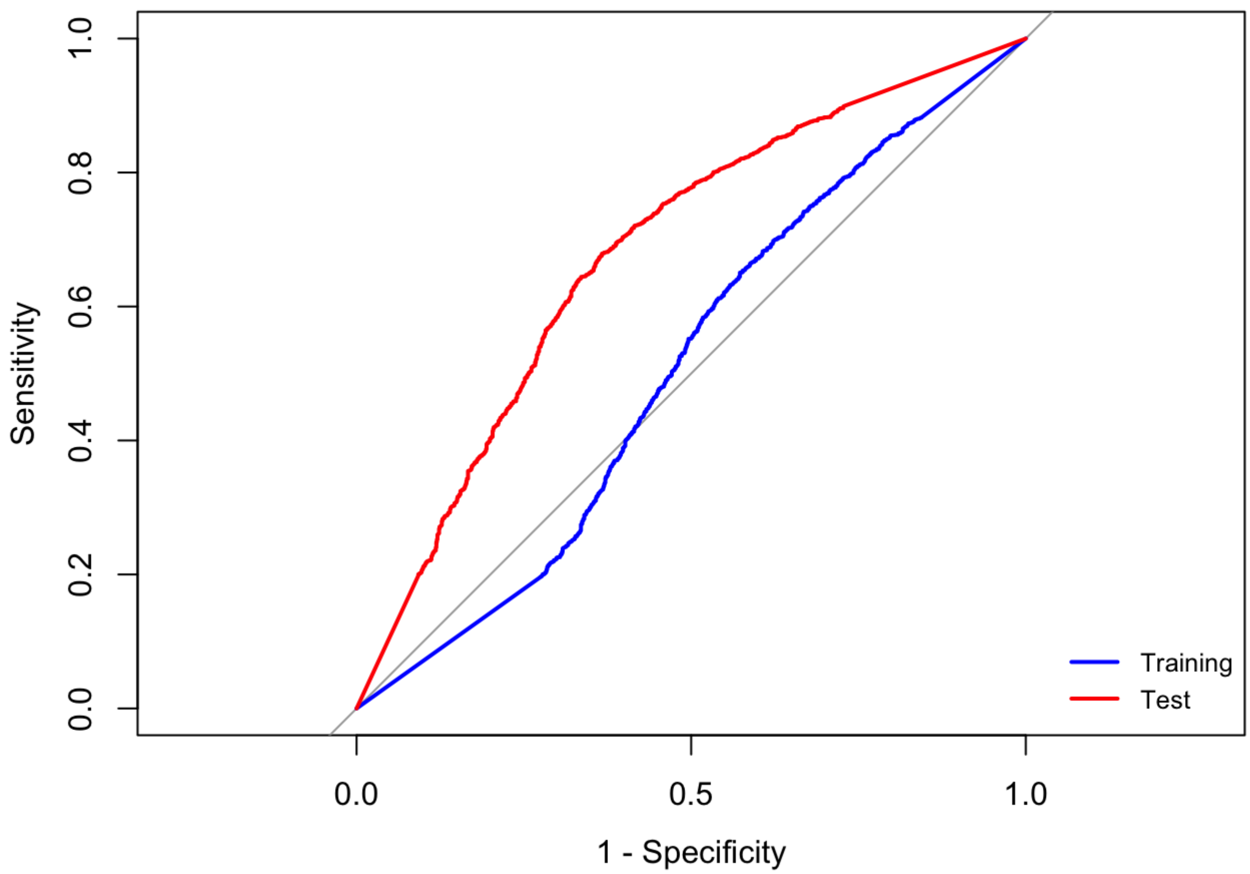
```
rocTst <- roc(revDTM_com_tst$hiLo, revSentiNdict_NBpredTst[,2], levels=c(-1, 1))
```

```
## Setting direction: controls < cases
```

```
bThr<-coords(rocTrn, "best", ret="threshold", transpose = FALSE)
bThr <- as.numeric(bThr)

bThr %>% view()

table(actual=revDTM_com_tst$hiLo, preds=revSentiNdict_NBpredTst[,2]>bThr)
```

```
##       preds
## actual FALSE TRUE
##     -1   759  466
##     1   1184 2591
```

```
plot.roc(rocTrn, col='blue', legacy.axes = TRUE)
plot.roc(rocTst, col='red', add=TRUE)
legend("bottomright", legend=c("Training", "Test"), col=c("blue", "red"), lwd=2, cex=0.8
, bty='n')
```



#SVM Model on Combined Dictioanry

```
#model 1
system.time(svmNDict1 <- svm(as.factor(hiLo) ~., data = revDTM_com_trn
%>% select(-review_id), kernel="radial", cost=1, gamma = 1,scale=FALSE, decision.values
 = TRUE))
```

```
##    user  system elapsed
##   4.317   0.640   6.407
```

```
revDTM_predTrn_svmNDict1<-predict(svmNDict1, revDTM_com_trn, decision.values = TRUE)
table(actual= revDTM_com_trn$hiLo, predicted= revDTM_predTrn_svmNDict1)
```

```
##       predicted
## actual   -1    1
##     -1  799  409
##     1    23 3769
```

```
revDTM_predTst_svmNDict1<-predict(svmNDict1, revDTM_com_tst, decision.values = TRUE)
table(actual= revDTM_com_tst$hiLo, predicted= revDTM_predTst_svmNDict1)
```

```
##       predicted
## actual   -1    1
##     -1  626  599
##     1    57 3718
```

```
auc(as.numeric(revDTM_com_trn$hiLo), as.numeric(revDTM_predTrn_svmNDict1))
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

```
## Area under the curve: 0.8277
```

```
auc(as.numeric(revDTM_com_tst$hiLo), as.numeric(revDTM_predTst_svmNDict1))
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.748
```

```
#model 2
system.time(svmNDict2 <- svm(as.factor(hiLo) ~., data = revDTM_com_trn
%>% select(-review_id), kernel="radial", cost=5, gamma = 5,scale=FALSE, decision.values
 = TRUE))
```

```
##    user  system elapsed
##   7.161   0.347   7.769
```

```
revDTM_predTrn_svmNDict2<-predict(svmNDict2, revDTM_com_trn, decision.values = TRUE)
table(actual= revDTM_com_trn$hiLo, predicted= revDTM_predTrn_svmNDict2)
```

```
##       predicted
## actual   -1    1
##     -1 1155   53
##     1     3 3789
```

```
revDTM_predTst_svmNDict2<-predict(svmNDict2, revDTM_com_tst, decision.values = TRUE)
table(actual= revDTM_com_tst$hiLo, predicted= revDTM_predTst_svmNDict2)
```

```
##       predicted
## actual   -1    1
##     -1  624  601
##     1   109 3666
```

```
auc(as.numeric(revDTM_com_trn$hiLo), as.numeric(revDTM_predTrn_svmNDict2))
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.9777
```

```
auc(as.numeric(revDTM_com_tst$hiLo), as.numeric(revDTM_predTst_svmNDict2))
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
## Area under the curve: 0.7403
```