Stats765: Milestone2

Shukri Shire -sshi482

15/04/2022

1. Goal The main goal of the group project is to analyse the deceptive reviews from the Deceptive Opinion Spam Corpus.

2. Data Source The main source of data came from Kaggle website using the decepive-opinion dataset.In order to get the dataset I had to make a free Kaggle account and download the data. The dataset provided was in a CSV format and I was able to download it into R.

The data consisted of reviews for the 20 most popular hotel reviews in Chicago and each hotel has 20 reviews consisting of either truthful or deceptive reviews. The original data provided was already well structured, as such each hotel has an equal number of truthful positive, deceptive positive,truthful negative, deceptive negative reviews. The whole dataset is used for analysis.

3. Data Processing The main aim of the data set is to wrangle the text data, and identify and similarities or differences between the types of deceptive reviews.

```
reviews = read_csv("deceptive-opinion.csv")
```

```
## Rows: 1600 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): deceptive, hotel, polarity, source, text
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dim(reviews)
```

```
## [1] 1600    5
```

```
names(reviews)
```

```
## [1] "deceptive" "hotel"     "polarity"  "source"    "text"
```

```
head(reviews)
```

```
## # A tibble: 6 x 5
##   deceptive hotel  polarity source      text
##   <chr>     <chr>  <chr>    <chr>       <chr>
## 1 truthful  conrad positive TripAdvisor "We stayed for a one night getaway with~
## 2 truthful  hyatt  positive TripAdvisor "Triple A rate with upgrade to view roo~
## 3 truthful  hyatt  positive TripAdvisor "This comes a little late as I'm finall~
## 4 truthful  omni   positive TripAdvisor "The Omni Chicago really delivers on al~
## 5 truthful  hyatt  positive TripAdvisor "I asked for a high floor away from the~
## 6 truthful  omni   positive TripAdvisor "I stayed at the Omni for one night fol~
```

A backup copy called Reviews2copy for data management purposes. The new dataset combined the columns deceptive and polarity to form a new column called group. The group column would allow us to see the

different combinations of deceptive and polarity, and we can use this as another layer of data classification. We chose to combine these columns because they are the best descriptors of a review's deceptive.

```
reviews2 <- reviews %>%
  unite(group, c(deceptive, polarity), remove = F) %>%
  mutate(review = row_number())

head(reviews2)
```

```
## # A tibble: 6 x 7
##   group              deceptive hotel  polarity source      text             review
##   <chr>              <chr>     <chr>  <chr>    <chr>       <chr>             <int>
## 1 truthful_positive truthful  conrad positive TripAdvisor "We stayed for~      1
## 2 truthful_positive truthful  hyatt  positive TripAdvisor "Triple A rate~      2
## 3 truthful_positive truthful  hyatt  positive TripAdvisor "This comes a ~      3
## 4 truthful_positive truthful  omni   positive TripAdvisor "The Omni Chic~      4
## 5 truthful_positive truthful  hyatt  positive TripAdvisor "I asked for a~      5
## 6 truthful_positive truthful  omni   positive TripAdvisor "I stayed at t~      6
```

Finally we used the tidy text, tidyverse packages in R to restructure, remove, and organised the data to show deceptive reviews.

4. Data Exploration We decided to focus on three main elements to help find deceptive reviews; 1. difference between neg and positive words 2. common words (e.g., word cloud) are deceptive and truthful reviews 3. total word count patterns between deceptive and positive.

We decided not to focus on source and hotel name unless further analysis requires identifying these columns. Additionally, the number of hotels and sources are divided between hotels, and for this reason, we decided to focus on the columns - deceptive, group, reviews, and polarity.

We decided to add new columns called polarity-score, reviews, positive words, and negative words to help measure/identify if a review is deceptive or not. We used the sentiment function in Tidytext to identify if words are positive or negative and score them using a polarity score.

As shown below, we have decided to use word clouds to see the most commonly used words and identify if there is any relationship between these words and positive, negative sentiment words. The word cloud function was ideal for providing a visual inspection of the commonly used words such as "I," "My," "Great," and "Comfortable". Also we found common words for positive and negative sentiments, for further analysis we could see how the frequency of these words are related to deceptive reviews in test and train data.

Other plots, such as box plots and bar plots, were used to identify relationships between the frequency of words, polarity, and sentiments. These plots show that there is not much difference between deceptive and truthful reviews in terms of word count and polarity score. We would need to find alternative parameters for identifying difference between deceptive and truthful reviews. –see next few pages for description of Data Exploration and analytical plan—

5. Analytical Plan

The next step is to identify which analytical methods are suitable for this data set. We already have a polarity score which means we can use this to provide more information about deceptive or non-deceptive reviews.

We identified that the word "I" , "my","we", "partner", and other sentimental are different between groups, so that we would look into this further for the next milestone.

A potential method we could employ includes LASSO, LDA, or Tree regression which are all classification methods. We have four groups of deceptive types, and we could use these methods to help bets classify reviews into different deceptive reviews.

Background:6:Appendix

```
stopwords <- as.list(get_stopwords()
                     [-c(1,2,3,4,5,6,7,8,58,63,65,67,69,73,75,79),-2])
#exclude personal pronouns - interested to see if fake reviews use less or more

stopwords <- as.data.frame(stopwords)

#remove stopwords
custom_stop_words <- tibble(word = c("hotel", "room", "chicago","chicago's"))
tidyreviews <- reviews2 %>%
  unnest_tokens(word, text) %>%
  anti_join(stopwords)%>%
  anti_join(custom_stop_words)
```

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

```
#negative/positive word count
bingnegative <- get_sentiments("bing") %>%
  filter(sentiment == "negative")

bingpositive <- get_sentiments("bing") %>%
  filter(sentiment == "positive")

negwords <- tidyreviews %>%
  semi_join(bingnegative, by = "word") %>%
  group_by(review) %>%
  summarise(negwords = n())

poswords <- tidyreviews %>%
  semi_join(bingpositive, by = "word") %>%
  group_by(review) %>%
  summarise(poswords = n())

tidyreviews <- tidyreviews %>%
  left_join(negwords, by = "review") %>%
  left_join(poswords, by = "review")

#include word count
tidyreviews <- tidyreviews %>%
  group_by(review) %>%
  summarize(wordcount = n()) %>%
  arrange(desc(wordcount)) %>%
  inner_join(tidyreviews,by = "review")


# Remove NA
names(which(colSums(is.na(tidyreviews))>0))
```

```
## [1] "negwords" "poswords"
```

```
head(tidyreviews$negwords %>% replace_na(0))
```

```
## [1] 26 26 26 26 26 26
```

```
head(tidyreviews$poswords %>%  replace_na(0))
```

```
## [1] 28 28 28 28 28 28
```

```
head(tidyreviews)
```

```
## # A tibble: 6 x 10
##   review wordcount group deceptive hotel polarity source word  negwords poswords
##    <int>     <int> <chr> <chr>     <chr> <chr>    <chr>  <chr>    <int>    <int>
## 1    999       412 trut~ truthful  talb~ negative Web    i           26       28
## 2    999       412 trut~ truthful  talb~ negative Web    much        26       28
## 3    999       412 trut~ truthful  talb~ negative Web    look~       26       28
## 4    999       412 trut~ truthful  talb~ negative Web    forw~       26       28
## 5    999       412 trut~ truthful  talb~ negative Web    our         26       28
## 6    999       412 trut~ truthful  talb~ negative Web    stay        26       28
```

4.Data Exploration

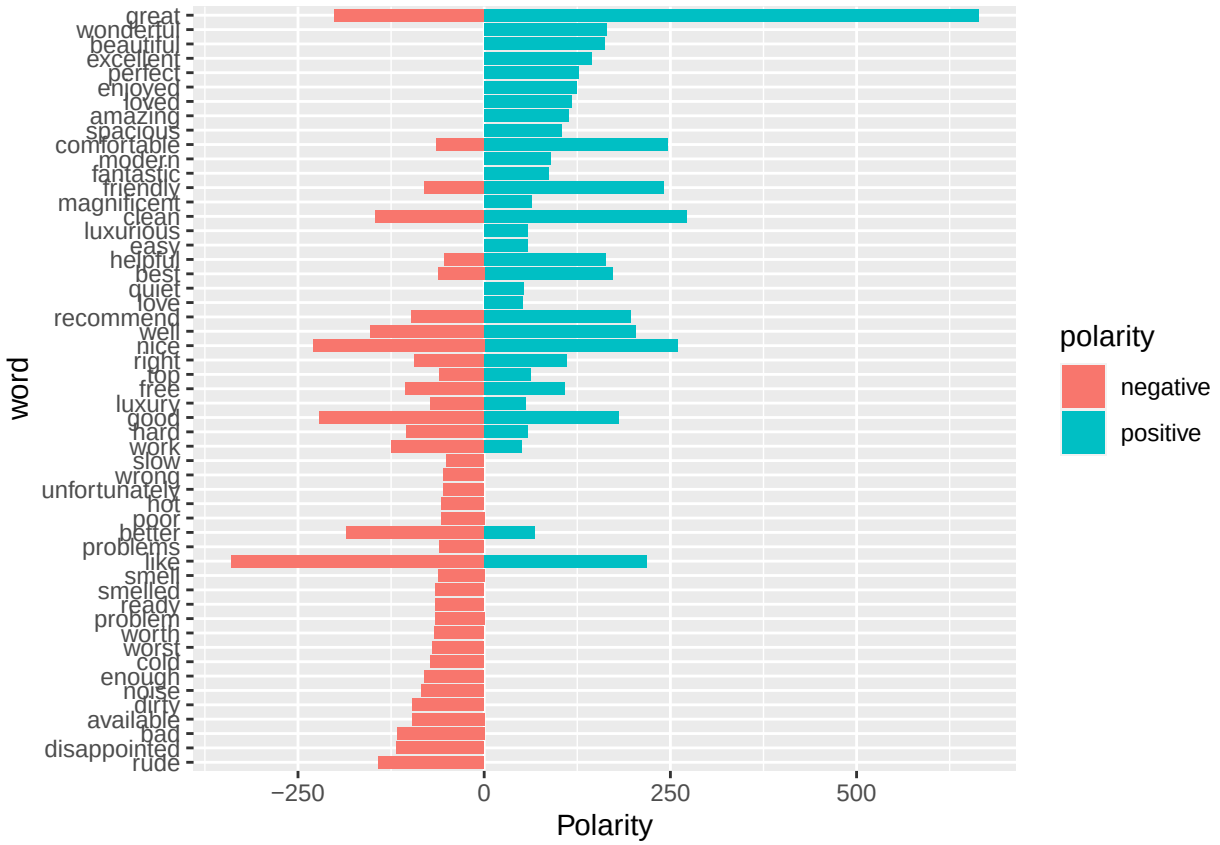
```
bing <- get_sentiments("bing")

tidyreviews <- tidyreviews %>%
  mutate(polarity_score=  poswords - negwords)
```

word cloud

```
word_count = tidyreviews %>%
  inner_join(bing) %>%
  count(word,polarity,sort = TRUE)
```

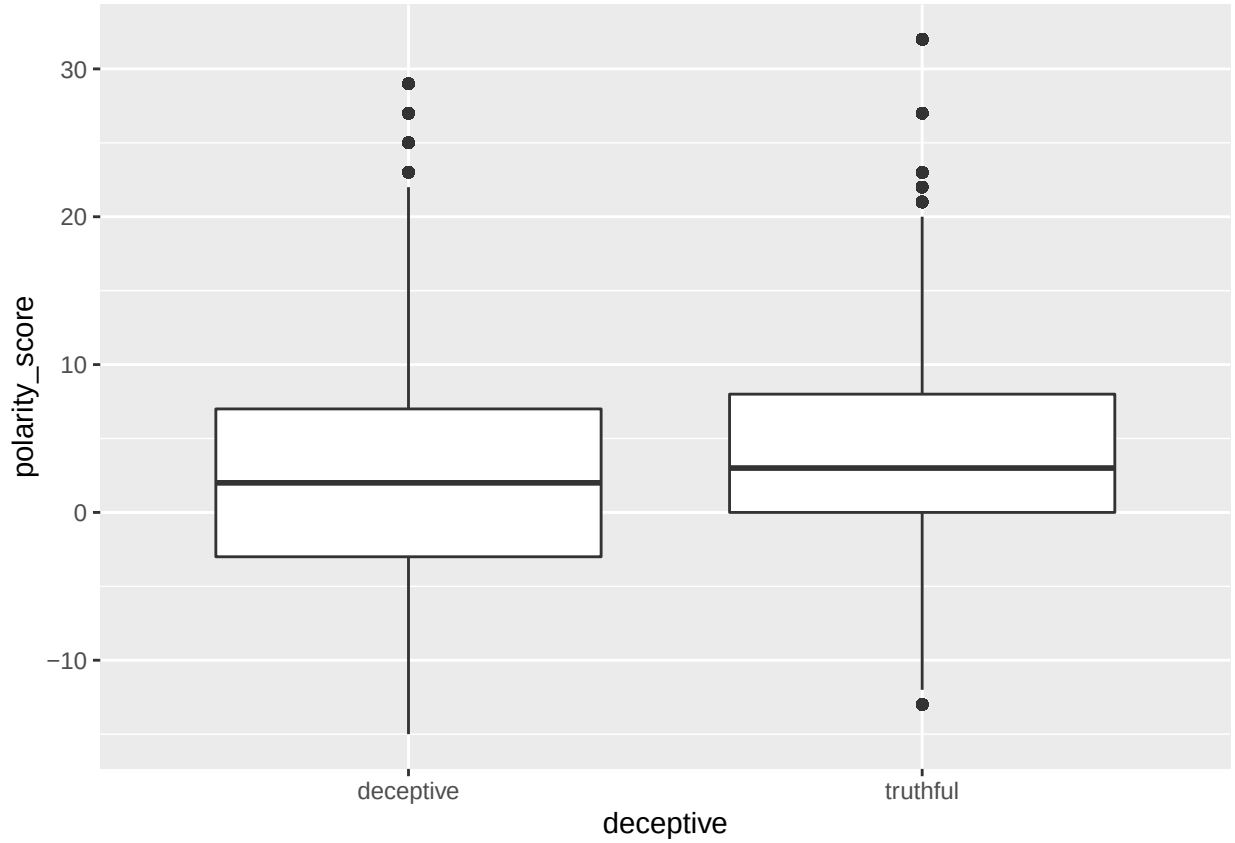
```
## Joining, by = "word"
```

```
  word_count%>%
  filter(n > 50) %>%
  mutate(n= ifelse(polarity == "negative",-n,n)) %>%
  mutate(word = reorder(word,n)) %>%
  ggplot(aes(word,n,fill=polarity))+
  geom_col()+
  coord_flip()+
  labs(y="Polarity")
```

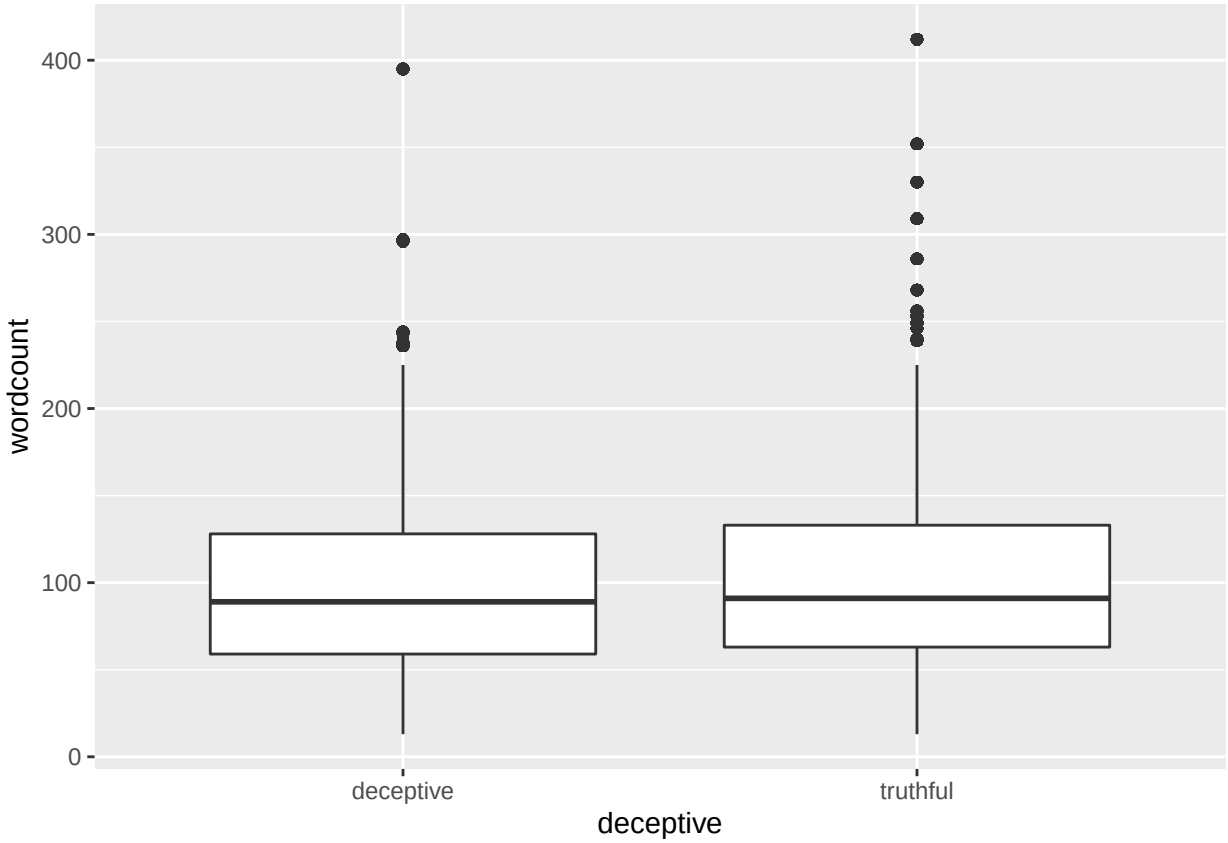



```
#1:difference between neg and positive words

ggplot(tidyreviews,aes(deceptive,polarity_score))+ geom_boxplot()
```

```
## Warning: Removed 20300 rows containing non-finite values (stat_boxplot).
```

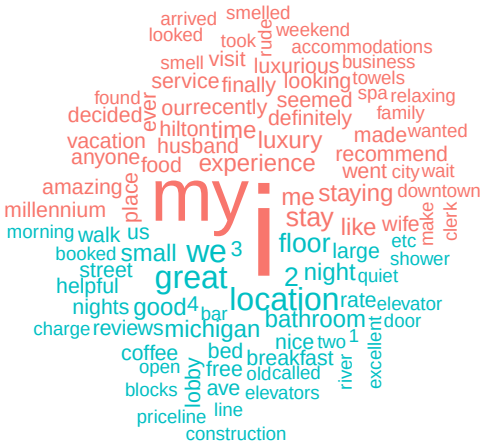



```
#3:Total word count patterns between deceptive and positive.

ggplot(tidyreviews, aes(deceptive,wordcount)) +geom_boxplot()
```



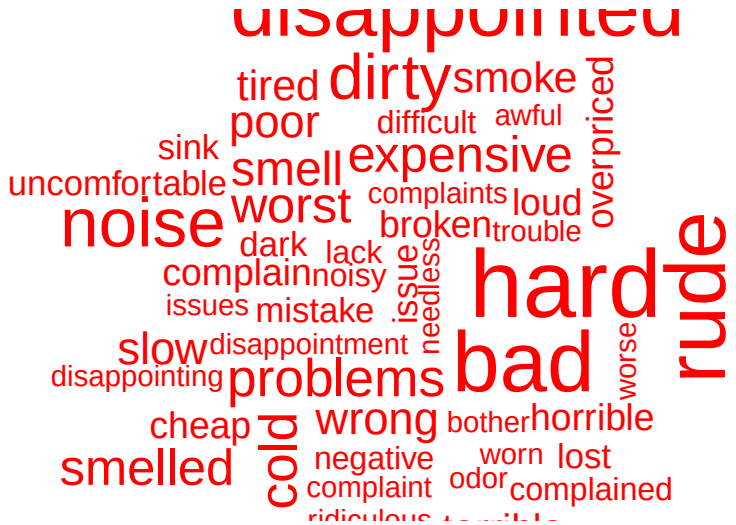

```
#2: Common words (e.g., word cloud) are deceptive and truthful reviews

#For all reviews
tidyreviews %>%
  count(word) %>%
  with(wordcloud(word,n,max.words = 50, random.order=F, col=rainbow(50),scale=c(8.5,0.75)))
```


```
#relationship between word and decptove coloumn
tidyreviews %>%
  count(word, deceptive,sort = TRUE) %>%
  acast(word ~deceptive, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("#F8766D", "#00BFC4"),
                   max.words = 100)
```




```
# Commonly used negative words
tidyreviews %>%
  inner_join(bingnegative) %>%
  count(word,sort = TRUE) %>%
  with(wordcloud(word,n,max.words = 50, colors = "red", scale=c(3,0.5)))
```

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




```
#Commonly used positive words
tidyreviews %>%
  inner_join(bingpositive) %>%
  count(word,sort = TRUE) %>%
  with(wordcloud(word,n,max.words = 50,colors = "green", scale=c(3.5,0.25)))
```

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


free
like
elegant
great
well
happy
helpful
convenient
enjoy
pleasant
cool
enough
complimentary
top
comfortable
impressed
fine
fantastic
spacious
right
quiet
enjoyed
better
pretty
love
excellent
worth
easy
clean
lovely
warm
perfect
beautiful
modern
fun
ready
friendly
good
best
loved
luxury
luxurious
magnificent
hot
nice
wonderful
available
amazing
work
recommend