Sentiment Analysis Project in R

Siddharth Prajapati

31st-July-2023

1) Introduction

Sentiment Analysis is a process of extracting opinions that have different polarities. By polarities, we mean positive, negative or neutral. It is also known as opinion mining and polarity detection. With the help of sentiment analysis, you can find out the nature of opinion that is reflected in documents, websites, social media feed, etc. Sentiment Analysis is a type of classification where the data is classified into different classes. These classes can be binary in nature (positive or negative) or, they can have multiple classes (happy, sad, angry, etc.).

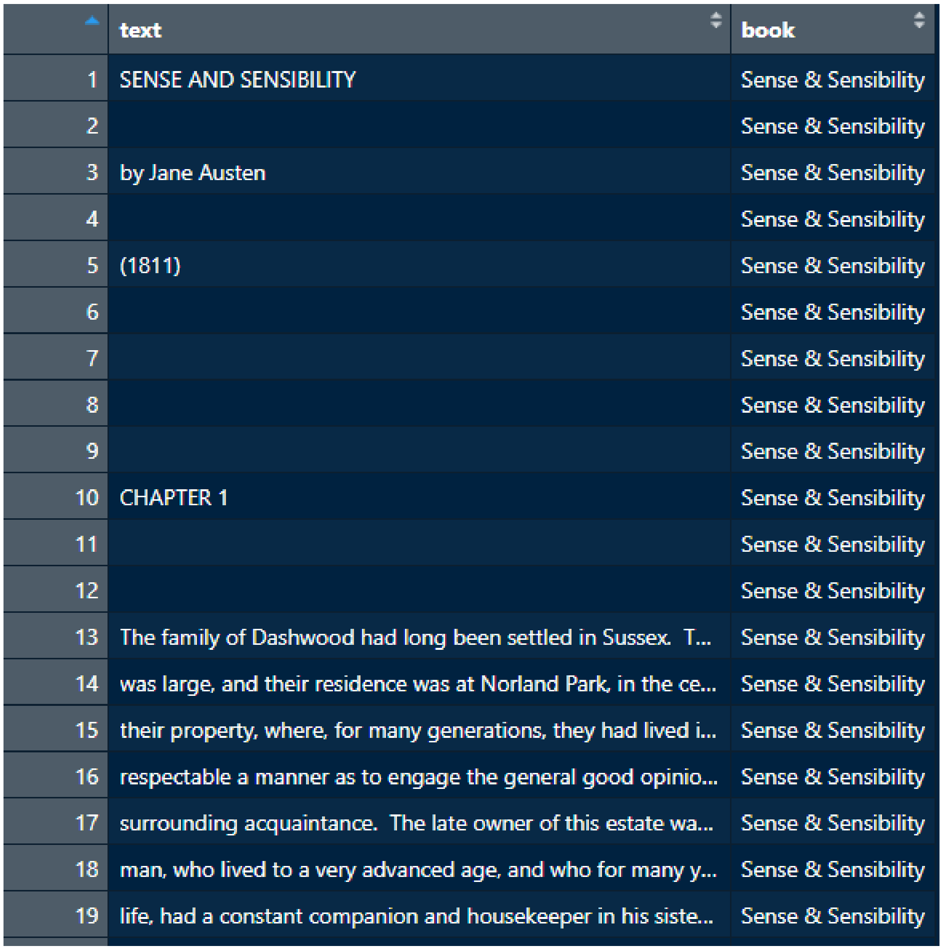
Important: The goal of this project is to practice the data cleaning, data manipulation & data visualization techniques using R.I do not claim copyright over any of the content here.

Source: https://data-flair.training/blogs/data-science-r-sentiment-analysis-project/

2) Understanding the data set

In this project, I have carry out sentiment analysis with R. The dataset that I have used here is provided by the R package 'janeaustenR'.

austen_books is a data frame of Jane Austen's 6 completed, published novels with two columns: **text**, which contains the text of the novels divided into elements of up to about 70 characters each, and **book**, which contains the titles of the novels as a factor in order of publication.



	text	book
1	SENSE AND SENSIBILITY	Sense & Sensibility
2		Sense & Sensibility
3	by Jane Austen	Sense & Sensibility
4		Sense & Sensibility
5	(1811)	Sense & Sensibility
6		Sense & Sensibility
7		Sense & Sensibility
8		Sense & Sensibility
9		Sense & Sensibility
10	CHAPTER 1	Sense & Sensibility
11		Sense & Sensibility
12		Sense & Sensibility
13	The family of Dashwood had long been settled in Sussex. T...	Sense & Sensibility
14	was large, and their residence was at Norland Park, in the ce...	Sense & Sensibility
15	their property, where, for many generations, they had lived i...	Sense & Sensibility
16	respectable a manner as to engage the general good opinio...	Sense & Sensibility
17	surrounding acquaintance. The late owner of this estate wa...	Sense & Sensibility
18	man, who lived to a very advanced age, and who for many y...	Sense & Sensibility
19	life, had a constant companion and housekeeper in his siste...	Sense & Sensibility

```
#load the dataset
library(janeaustenr)

# get the summary of dataset
summary(austen_books())
```

```
##      text                                book
##  Length:73422       Sense & Sensibility:12624
##  Class :character   Pride & Prejudice  :13030
##  Mode  :character   Mansfield Park     :15349
##                     Emma               :16235
##                     Northanger Abbey   : 7856
##                     Persuasion         : 8328
```

3) Developing Sentiment Analysis Model in R

In order to build the project on sentiment analysis, I had use of the *tidytext* package that comprises of sentiment lexicons that are present in the dataset of *sentiments*.

A sentiment lexicon is a collection of words (also known as polar or opinion words) associated with their sentiment orientation, that is, positive or negative.

In this project, I have use the **bing lexicons** to extract the sentiments out of our data.

```r
# load the dataset
library(tidytext)

# view the bing sentiments data
get_sentiments("bing")
```

```
## # A tibble: 6,786 x 2
##    word        sentiment
##    <chr>       <chr>
##  1 2-faces     negative
##  2 abnormal    negative
##  3 abolish     negative
##  4 abominable  negative
##  5 abominably  negative
##  6 abominate   negative
##  7 abomination negative
##  8 abort       negative
##  9 aborted     negative
## 10 aborts      negative
## # i 6,776 more rows
```

```r
tail(get_sentiments("bing"))
```

```
## # A tibble: 6 x 2
##   word      sentiment
##   <chr>     <chr>
## 1 zealous   negative
## 2 zealously negative
## 3 zenith    positive
## 4 zest      positive
## 5 zippy     positive
## 6 zombie    negative
```

4) Data Wrangling

In this step, I have used the data wrangling techniques to prepare the data for further analysis.

The *austen_books* data will provide the text of the novel in one column which is *text*.

Tidytext package helped me to perform efficient text analysis on our data. I have convert the text of our books into a tidy format using "unnest_tokens()" function.

tidyverse package helped me to do the data processing efficiently.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
tidy_data <- austen_books() %>%
             group_by(book) %>%
             mutate(linenumber = row_number(),
             chapter = cumsum(str_detect(text,regex("^chapter [\\divxlc]",
                                                           ignore_case=TRUE)))) %>%
             ungroup() %>%
             unnest_tokens(word,text)
head(tidy_data,10)
```

```
## # A tibble: 10 x 4
##    book                linenumber chapter word
##    <fct>                    <int>   <int> <chr>
##  1 Sense & Sensibility          1       0 sense
##  2 Sense & Sensibility          1       0 and
##  3 Sense & Sensibility          1       0 sensibility
##  4 Sense & Sensibility          3       0 by
##  5 Sense & Sensibility          3       0 jane
##  6 Sense & Sensibility          3       0 austen
##  7 Sense & Sensibility          5       0 1811
##  8 Sense & Sensibility         10       1 chapter
##  9 Sense & Sensibility         10       1 1
## 10 Sense & Sensibility         13       1 the
```

Understanding the above code

group_by(book) %>%
This line groups the data by the book column, which means that subsequent operations will be applied separately to each book in the dataset.

mutate(linenumber = row_number() , This line adds a new column called linenumber to the dataset. The row_number() function generates a sequential number for each row, which will represent the line number of the text.

chapter = cumsum(str_detect(text,regex(“ˆchapter [\divxlc]”, ignore_case=TRUE)))) %>%
This line creates another new column called chapter. It uses the str_detect function with a regular expression to identify lines that start with “chapter” (ignoring case). The cumsum function then calculates the cumulative sum of the results (1 for lines starting with “chapter” and 0 for others). This creates a sequence that increments whenever a new chapter begins.

ungroup() %>%
This line ungroups the data, which means that subsequent operations will be applied to the entire dataset as a whole, rather than by individual book groups.

unnest_tokens(word,text)
This line tokenizes the text column, which means it breaks down the text into individual words and stores them in a new column called word. This process effectively separates the text into a tidy format where each row represents a single word.

In summary, the code takes a dataset containing the text of various books by Jane Austen, groups it by book, adds columns for line numbers and chapters, ungroups the data, and finally tokenizes the text into individual words, resulting in a tidy dataset with one word per row along with information about the line number and the chapter the word belongs to.

After that I have proceed towards counting the most common positive and negative words that are present in the tidy_data.

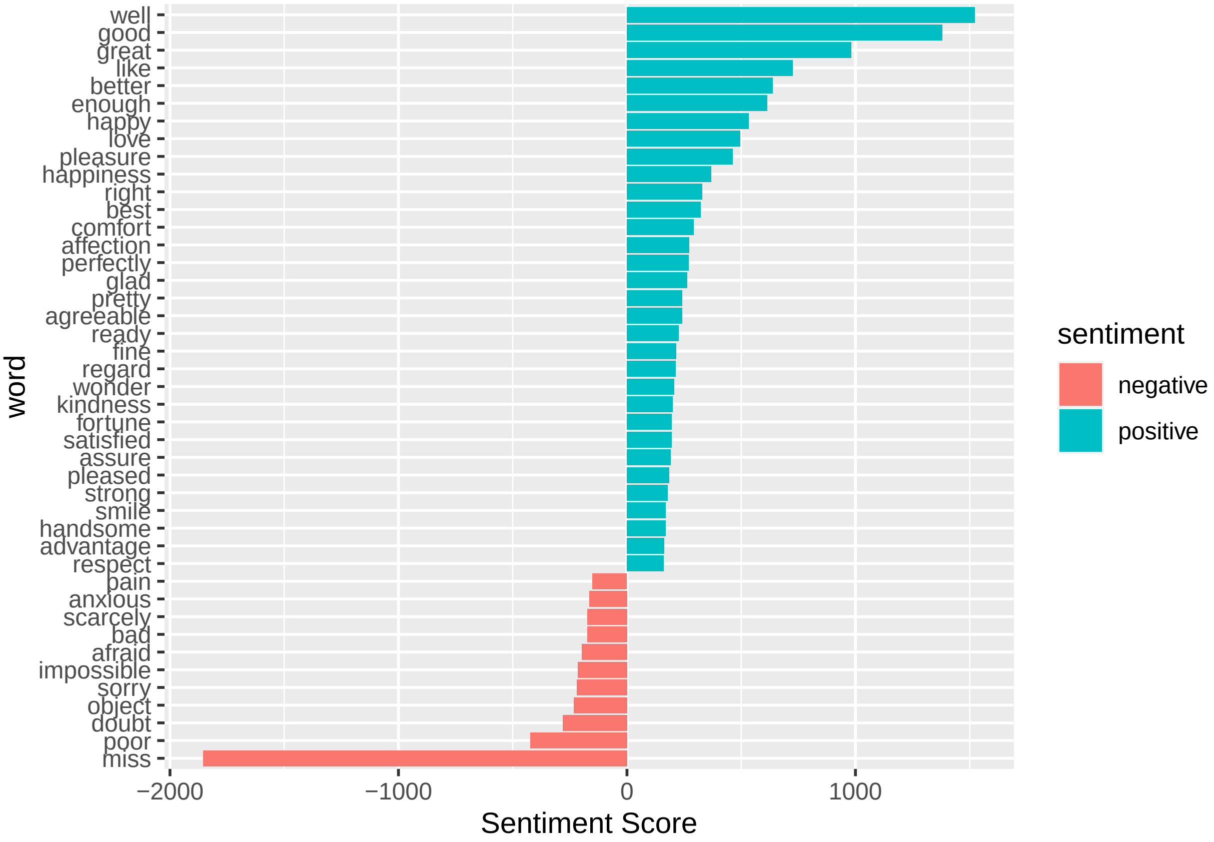
```r
bing <- get_sentiments("bing")

counting_words <- tidy_data %>%
                  inner_join(bing) %>%
                  count(word, sentiment, sort = TRUE)
head(counting_words)
```

```
## # A tibble: 6 x 3
##   word   sentiment     n
##   <chr>  <chr>     <int>
## 1 miss   negative   1855
## 2 well   positive   1523
## 3 good   positive   1380
## 4 great  positive    981
## 5 like   positive    725
## 6 better positive    639
```

4) Data Visualization

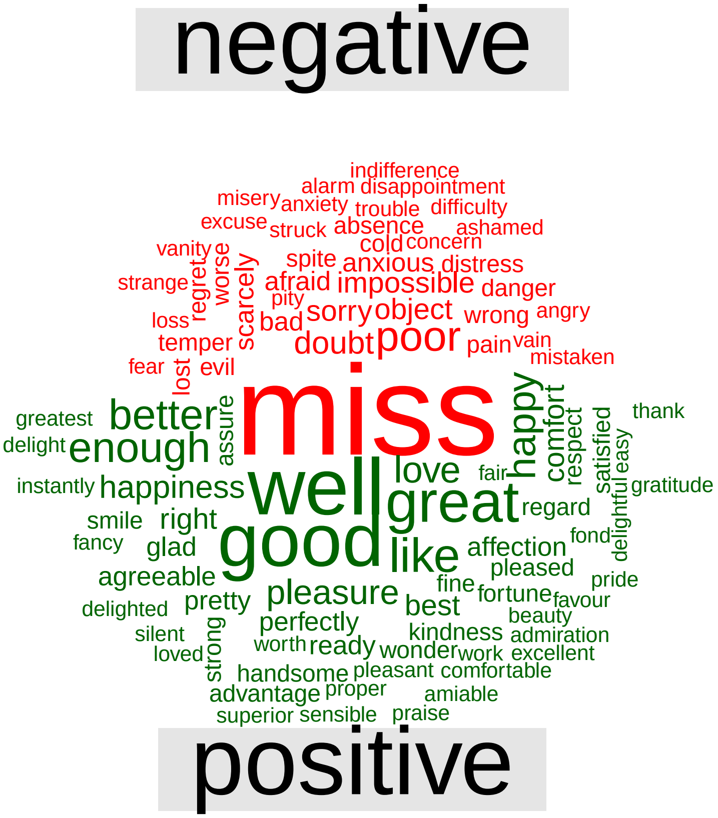
In the next step, we I have perform visualization of our sentiment score. I have plot the scores along the axis that is labeled with both positive as well as negative words. I have use ggplot() function to visualize our data based on their scores.

```r
counting_words %>%
 filter(n > 150) %>%
 mutate(n = ifelse(sentiment == "negative", -n, n)) %>%
 mutate(word = reorder(word, n)) %>%
 ggplot(aes(word, n, fill = sentiment))+
 geom_col() +
 coord_flip() +
 labs(y = "Sentiment Score")
```



In the final visualization, I had created a wordcloud that will describe the most recurring positive and negative words. In particular, I have use the 'comparision.cloud()' function to plot both negative and positive words in a single wordcloud as follows:

```
library(reshape2)
library(wordcloud)
tidy_data %>%
 inner_join(bing) %>%
 count(word, sentiment, sort = TRUE) %>%
 acast(word ~ sentiment, value.var = "n", fill = 0) %>%
 comparison.cloud(colors = c("red", "dark green"),
          max.words = 100)
```



5) Summary

- In this project, I have learnt about the concept of sentiment analysis and implemented it over the dataset of Jane Austen's books.
- first I have perform the data wrangling on the given dataset to make the data in the proper format for analysis.
- After that, create the different visualizations on the sentiment scores to present the findings in the more compelling way.