Project -1

Problem 4 – Tweet Analysis for Real Direct

CSE – 587

Instructor – Bina Ramammurthy

Submitted By:-

NALIN KUMAR

Person - 50170479

The problem statement asked us to collect tweets for around a week and perform tweet analysis using some statistical analysis in order to judge whether apartment rental would be a fruitful business for Real Direct or not. In order to achieve this, I collected tweets for a week and analysed the text part of the tweets based on finding out the most frequent occurring terms and various correlations between terms occurring in the tweets. For this, I used various packages and their respective functions and the most important of these are as follows:-

1. **Package tm** - The main structure for managing documents in tm is a so-called Corpus, representing a collection of text documents. A corpus is an abstract concept, and there can exist several implementations in parallel. The default implementation is the so-called VCorpus (short for Volatile Corpus) which realizes a semantics as known from most R objects: corpora are R objects held fully in memory. We denote this as volatile since once the R object is destroyed, the whole corpus is gone. This package has several built in functions like tm_map which further cleans up the text extracted from the weekly tweets based on lowercase, removing punctuation, special characters, whitespaces, stopwords etc. I also used the TermDocumentMatrix function to create a matrix of terms in the first column and their respective occurences in all the tweets into other columns. I then simply performed a sum on all the columns for all the rows using the rowSum function. Finally, I constructed a data frame object which contains terms and their respective frequencies in a table like structure. This term document matrix and data frame object can then be passed on to other functions to perform a meaningful analysis for the same like findAssocs(), findFrequentTrems() etc. which finds out the associations in a term document matrix and finds out the most frequent terms respectively
2. **Package wordCloud** – I simply used this package to create a word cloud of most frequent terms (generally having term frequency > 50). I used this because the relative occurrences of most frequent terms can be clearly visualized based on the size of the picture of that word inside the word cloud generated image

I performed tweet analysis in the following steps:-

- Initially I loaded the required libraries as follows
  library("twitteR")
  library("wordcloud")
  library("tm")
  library("ggplot2")
  library("Rgraphviz")

- Then, I read the data from json file containing real estate and rental tweets and converted those tweets into a data frame object. Subsequent to this, I extracted the text portion from all of the tweets as shown in the following code snippet

  json_data1 <- readLines('Mar1001.json', warn = FALSE)
  json_df <- jsonlite::fromJSON(json_data1)
  tweets_text <- json_df$text

- Next, I removed unnecessary characters from the text portion of the tweets since certain tweets could not be parsed due to the presence of these characters and gives

errors. For this, I used iconv() function which essentially converts string to requested character encoding

rent_tweets_text <- iconv(tweets_text, 'UTF-8', 'ASCII')

- Subsequent to this, I created a corpus which is essentially a character vector and then cleaned up the tweets text using tm_map() function from tm package based on stopwords, punctuation, whitespaces and by specifying regex pattern for other unnecessary characters like http, https etc. The code snippet is as follows

```
rent_clean_text <- Corpus(VectorSource(rent_tweets_text))
rent_clean_text <- tm_map(rent_clean_text, content_transformer(function (x , pattern ) gsub(pattern, " ", x)), "/")
rent_clean_text <- tm_map(rent_clean_text, content_transformer(function (x , pattern ) gsub(pattern, " ", x)), "@")
rent_clean_text <- tm_map(rent_clean_text, content_transformer(function (x , pattern ) gsub(pattern, " ", x)), "\\|")
rent_clean_text <- tm_map(rent_clean_text, removePunctuation)
rent_clean_text <- tm_map(rent_clean_text, content_transformer(tolower))
rent_clean_text <- tm_map(rent_clean_text, removeWords, stopwords("english"))
rent_clean_text <- tm_map(rent_clean_text, stripWhitespace)
rent_clean_text <- tm_map(rent_clean_text, removeWords, c("http","https", "tco"))
```

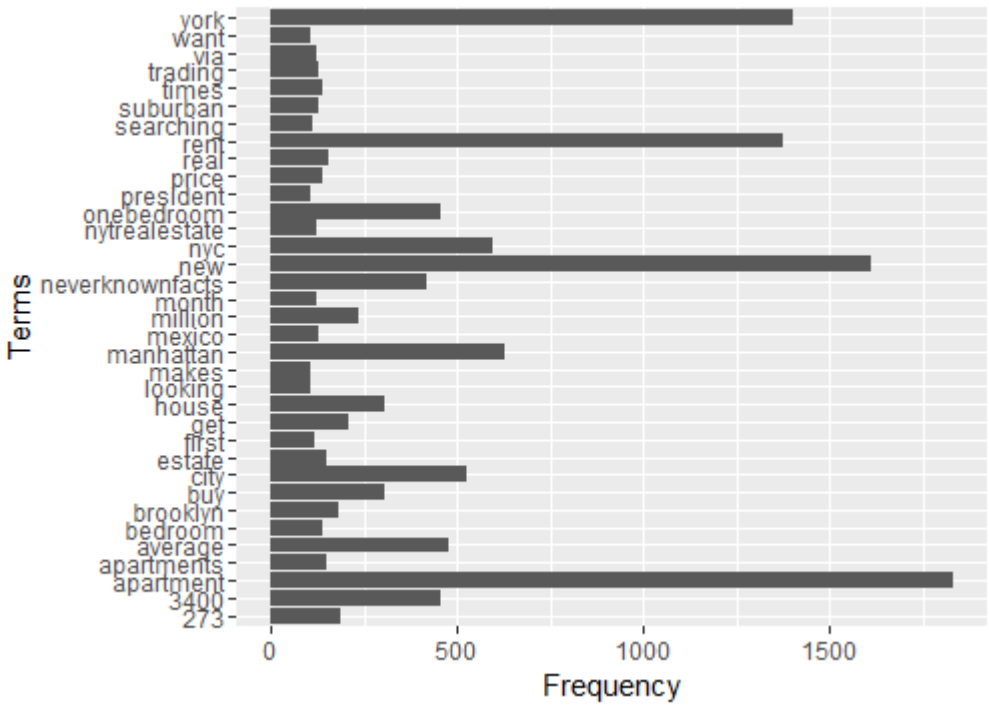
- In the next step, I constructed a term document matrix which contains terms in the first column and their respective occurrences in all the tweets in all other columns for a particular term. Then, I performed a sum over all the columns for each individual row which actually gave me the term frequency for those terms and sorted them in decreasing order so that the most frequent terms appear at the top. Also, I created a data frame object which contains a mapping of all terms and their respective term frequencies stored in a table like structure

```
tdm <- TermDocumentMatrix(rent_clean_text)
tmatrix <- as.matrix(tdm)
term_freqs <- sort(rowSums(tmatrix),decreasing=TRUE)
term_freqs <- subset(term_freqs, term_freqs >= 100)
termfreq_df <- data.frame(term = names(term_freqs),freq=term_freqs)
termfreq_df[1:10, ]
```

```
> termfreq_df[1:10, ]
                 term freq
apartment   apartment 1832
new               new 1610
york             york 1402
rent             rent 1374
manhattan   manhattan  626
nyc               nyc  593
city             city  526
average       average  479
3400             3400  454
onebedroom onebedroom  454
>
```

- In the next step I constructed a ggplot using the ggplot2 package where most frequent terms were plotted on the y-axis and their respective frequencies on the x-axis. In order to fit all the most relevant frequent terms in this single graph, I plotted only those term frequencies which were greater than 100 in my case (since there were a significant number of terms having term frequency greater than 100) and worked on this subset of data. As can be seen in the plot below, there are some prominent terms with term frequencies even more than 300 like York, rent, new, apartment, nyc, onebedroom, city, average, 3400, manhattan etc. This gives us some kind of idea that most of the people wrote about that they were looking a one bedroom apartment in manhattan and the average monthly price for that apartment is 3400$. This also tells us the fact since most of the people tweeted about a one bedroom apartment, that means they will most probably cannot afford or want to buy an apartment in one of the most expensive areas on new York like manhattan

ggplot(termfreq_df, aes(x = term, y = freq)) + geom_bar(stat = "identity") + xlab("Terms") + ylab("Frequency") + coord_flip()



- In the next step, I created a word cloud using wordcloud package which will tell me the relative occurrence of various terms in a single image as shown below. The following image clearly highlights the fact that people tweeted about both apartment rental as well as real esatate(as can be seen in the word nytrealestate below). But, the difference being that the words depicting that the average monthly rent for a one bedroom apartment in manhattan new York is $3400 are much larger in size as compared as compared to words like buy and real estate. Also, we can draw one more important observation that most of people searched for renting apartments in manhattan than Brooklyn due to the relative size of images of words manhattan and brookyln. Hence, wordcloud somehow helped me to extend my analysis to a next level and confirm my assumptions from the previous ggplot

  wordcloud(words = termfreq_df$term, freq = termfreq_df$freq, min.freq = 1,
  max.words=200, random.order=FALSE, rot.per=0.35,
  colors=brewer.pal(8, "Dark2"))

- In the next part, I tried using some functions provided by tm package to find associations and frequent terms like findAssocs(), findFreqTerms(). As can be seen below, the result of running findAssocs() functions clearly shows that rent keyword has the strongest correlation with other words like apartment, 3400, average, onebedroom, neverknownfacts, manhattan, nyc, bedroom, new, searching, financial, district as we deduced in previous parts.

  findAssocs(tdm, "rent", 0.2)

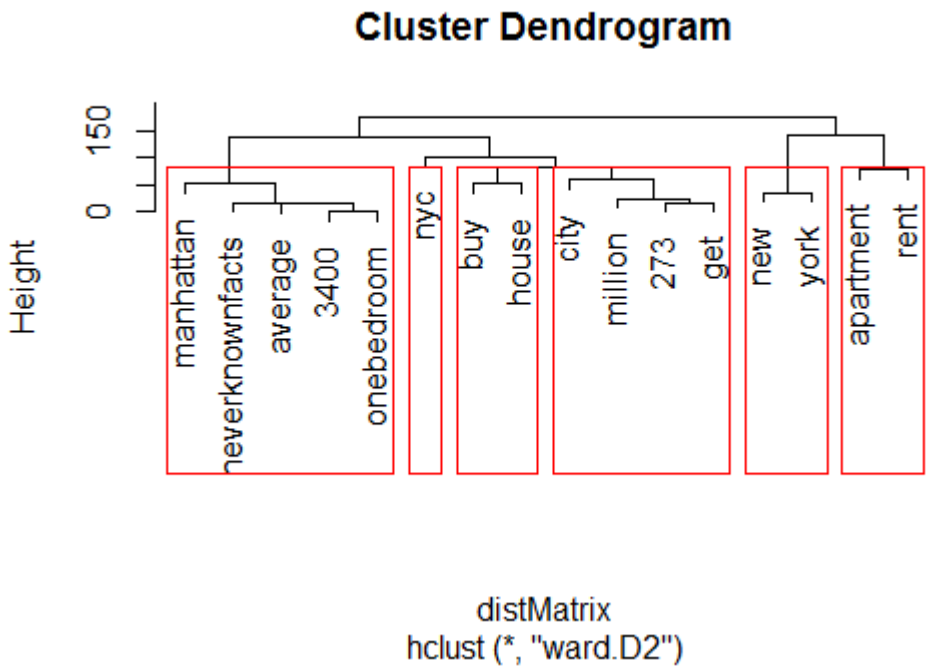
```
> findAssocs(tdm, "rent", 0.2)
$rent
      apartment            3400         average      onebedroom neverknownfacts       manhattan
           0.66            0.47            0.47            0.47            0.45            0.43
            nyc         bedroom             new       searching       financial        district
           0.26            0.24            0.22            0.22            0.21            0.20

>
```

frequent_terms <- findFreqTerms(tdm, lowfreq = 25)
print(frequent_terms)

```
Console ~/
> frequent_terms <- findFreqTerms(tdm, lowfreq = 25)
> print(frequent_terms)
  [1] "1200"                "1bedroom"            "273"
  [4] "3400"                "500k"                "700"
  [7] "aguascalientes"      "america"             "amp"
 [10] "apartment"           "apartments"          "apartmentscom"
 [13] "average"             "bedroom"             "brooklyn"
 [16] "businessinsider"     "buy"                 "buying"
 [19] "call"                "campus"              "can"
 [22] "cant"                "celebrity"           "city"
 [25] "cohen"               "controlled"          "credit"
 [28] "cruz"                "district"            "dnainfo"
 [31] "dumbo"               "east"                "edu00a0u00bdedu00b2u0080"
 [34] "election"            "engine"              "estate"
 [37] "even"                "financial"           "first"
```

- In the final step, I created a cluster dendrogram where initially I removed all the sparse terms and then found out a final correlation amongst the most frequent terms by visualizing this through a graph. As can be seen in the dendrogram below, (rent, apartment) and (new, york) are most similar to each other. In an analogous manner, (buy, house) and (manhattan, average, 3400, onebedroom)are similar to each other. If we go one level above, we can make one more interesting observation that (nyc,city,273,million,buy, house) all lie in the same level. This draws another conclusion that the cost of buying a house in nyc is around 273 million dollars in addition to the conclusion that the average monthly rent of apartment in manhattan new York is $3400.



Conclusions (Pricing Model and Subscription)

Based on all the analysis above, we can conclude that most of the tweets comprised of apartment rental in New York as compared to real estate in new York. This proves that most of the people are willing to rent an apartment in New York rather than buying a new one. Also, we analysed that most of the people were willing to rent an apartment in Manhattan as opposed to Brooklyn. In addition to this, we also came to the conclusion that the average monthly rent of apartment in Manhattan region of New York is $3400 and the average cost of buying a house in NYC is $273 million. Based on this vital pricing information, Real Direct can fruitfully provide apartment rental as a service. Looking at the user trends, we need to be more generous towards people who are willing to rent an apartment rather than the ones who want to buy an apartment since their number is much larger than the latter ones and based on this hypothesis we can provide recommendations to Real Direct to devise their pricing model and subscription accordingly. The pricing model which I would suggest for apartment rental is that we will provide some basic services free of cost while we will be charging the users with a one-time nominal fee if they subscribe to our advanced services such as a personal agent. This one time nominal should be proportional to the average monthly apartment rent which in our case is $3400. Whereas, for the buying model, I would suggest levying a fee of 1% of the total cost of apartment every time the user buys a new house and avails our agent services since the frequency of buying of houses by the users is much lesser than the frequency of user who will be willing to rent an apartment through Real Direct. Henceforth, our focus group will be those users who wish to stay in a rented apartment and our policies should be flexible to this particular user group. This might eventually lead to the success of introducing Apartment rental as a service by Real Direct in terms of profitability and consumer base of the organization.

References

- http://rstudio-pubs-static.s3.amazonaws.com/82966_c8fee3e3107241678aa6ecebdd831a41.html
- http://www2.rdatamining.com/uploads/5/7/1/3/57136767/rdatamining-slides-text-mining.pdf
- https://sites.google.com/site/miningtwitter/basics/text-mining
- http://www.unt.edu/rss/class/Jon/Benchmarks/TextMining_L_JDS_Jan2014.pdf
- https://cran.r-project.org/web/packages/tm/tm.pdf