PHASE 4 -PROJECT

Title: Continuing to Build Your Sentiment Analysis Solution with NLP Techniques

Introduction:

Sentiment analysis has become a crucial tool for understanding customer feedback, social media trends, and public opinion. In the pursuit of creating a robust sentiment analysis solution, incorporating Natural Language Processing (NLP) techniques is essential. NLP allows you to extract valuable insights from textual data, enabling more accurate and insightful sentiment analysis. In this article, we will explore how to continue building your sentiment analysis solution by employing NLP techniques.

1. Data Preprocessing:

To build an effective sentiment analysis model, start by preparing your data. Data preprocessing is a critical step that involves tasks such as text tokenization, lowercasing, and removing stop words and punctuation. NLP libraries like NLTK (Natural Language Toolkit) and spaCy are invaluable for these tasks. Tokenization breaks text into words or phrases, which is essential for subsequent analysis.

2. Feature Extraction:

Feature extraction is pivotal in NLP-based sentiment analysis. Convert text data into numerical features that your model can understand. Techniques like TF-IDF (Term Frequency-Inverse Document Frequency) or Word Embeddings (e.g., Word2Vec, GloVe) are commonly used for this purpose. These representations capture the semantic meaning of words and help your model better understand context.

3. Sentiment Lexicons:

Sentiment lexicons are curated lists of words and phrases with associated sentiment scores (positive, negative, neutral). By utilizing sentiment lexicons, you can assess the sentiment of a given text based on the words it contains. Popular lexicons like AFINN, VADER, and SentiWordNet can be integrated into your sentiment analysis solution.

4. Machine Learning Models:

Incorporate machine learning models to classify sentiments. Common choices include Naive Bayes, Support Vector Machines (SVM), and deep learning models like Recurrent Neural Networks (RNNs) or Transformers. Training these models on labeled sentiment data is a crucial step. Tools like scikit-learn and TensorFlow can be used for model development.

5. Aspect-Based Sentiment Analysis:

Moving beyond document-level sentiment analysis, aspect-based sentiment analysis allows you to analyze sentiment at a more granular level. It identifies and evaluates sentiment towards specific aspects or entities mentioned in the text, providing deeper insights.

6. Sentiment Visualization:

Visualizing sentiment analysis results can make the data more accessible and actionable. Create interactive dashboards or visual representations to present the sentiment trends over time or across different sources.

7. Continuous Learning:

NLP and sentiment analysis are dynamic fields. Stay updated with the latest developments, tools, and datasets. Continuously fine-tune your models and improve accuracy. Explore pre-trained NLP models like BERT and GPT-3 for even more advanced sentiment analysis.

8. Quality Data:

The quality of your training data plays a pivotal role in the success of your sentiment analysis solution. Make sure your data is clean, diverse, and accurately labeled. Data augmentation and crowdsourcing can help enhance the quality of your dataset.

9. Error Analysis:

Regularly conduct error analysis to understand where your model might be failing. This will help you fine-tune the model and make necessary improvements to increase accuracy.

Exploring the Airline Twitter Sentiment Data

This data was originally posted by Crowdflower last February and includes tweets about 6 major US airlines. Additionally, Crowdflower had their workers extract the sentiment from the tweet as well as what the passenger was dissapointed about if the tweet was negative.

I've done some minimal preprocessing on this data and re-released it on Kaggle as a CSV file and SQLite database.

```
library(RSQLite)

db <- dbConnect(dbDriver("SQLite"), "../input/database.sqlite")
```

First, let's see what tables we have to work with.

```
library(dplyr)

tables <- dbGetQuery(db, "SELECT Name FROM sqlite_master WHERE type='table
'")

colnames(tables) <- c("Name")

tables <- tables %>%

          rowwise() %>%

          mutate(RowCount=dbGetQuery(db, paste0("SELECT COUNT(*) RowCount
FROM ", Name))$RowCount[1])

print.table(tables)
```

Name	RowCount
Tweets	14485

As we see above, there's a single table: Tweets. Now let's see what this table contains.

```
print.table(dbGetQuery(db, "

SELECT *

FROM Tweets

LIMIT 6"))
```

We see that, in addition to the raw tweets and some standard data about that, Crowdflower's extracted the airline the tweet's about as well as the sentiment and the reason the tweet was negative (if it was negative).

Let's see how often airlines are mentioned and what the sentiment tends to look like.

```
Library

(ggvis)

dbGetQuery(db, "

SELECT airline Airline,

       airline_sentiment Sentiment,

       COUNT(airline) NumTweets

FROM Tweets

GROUP BY airline,

         airline_sentiment") %>%

  ggvis(~Airline, ~NumTweets, fill=~Sentiment) %>%

  layer_bars()
```

American Delta South west United US Airways Virgin AmericaAirline05001,0001,5002,0002,5003,0003,5004,000NumTweetsSentimentnegativeneutralpositive

From this, we see that United had the most twitter commentary on it and US Airways had the highest fraction of negative twitter commentary. Virgin America had the least twitter commentary, but it also had the highest fraction of positive commentary.

One question that comes to mind: when tweets are negative, why are they negative mind?

```
print.table(dbGetQuery(db, "

SELECT airline,

       negativereason,

       COUNT(negativereason)

FROM Tweets

GROUP BY airline,

         negativereason

ORDER BY COUNT(negativereason) DESC"))
```

airline	negativereason	COUNT(negativereason)
Delta		1267
Southwest		1234
United		1189
US Airways	Customer Service Issue	811
American	Customer Service Issue	743
American		740
United	Customer Service Issue	681
US Airways		650
United	Late Flight	525
US Airways	Late Flight	453
Southwest	Customer Service Issue	391

United	Can’t Tell	379
Virgin America		323
Delta	Late Flight	269
United	Lost Luggage	269
US Airways	Can’t Tell	246
American	Late Flight	234
American	Cancelled Flight	228
United	Bad Flight	216
Delta	Customer Service Issue	199
US Airways	Cancelled Flight	189
Delta	Can’t Tell	186
American	Can’t Tell	184
United	Cancelled Flight	181

United	Flight Attendant Complaints	168
Southwest	Cancelled Flight	162
Southwest	Can’t Tell	159
US Airways	Lost Luggage	154
Southwest	Late Flight	152
American	Lost Luggage	144
United	Flight Booking Problems	144
American	Flight Booking Problems	124
US Airways	Flight Attendant Complaints	123
US Airways	Flight Booking Problems	122
US Airways	Bad Flight	104
Southwest	Bad Flight	90
Southwest	Lost Luggage	90

American	Bad Flight	82
American	Flight Attendant Complaints	81
Delta	Bad Flight	64
Southwest	Flight Booking Problems	61
Delta	Flight Attendant Complaints	60
Virgin America	Customer Service Issue	60
Delta	Lost Luggage	57
Delta	Cancelled Flight	51
US Airways	longlines	50
United	longlines	48
Delta	Flight Booking Problems	44
Southwest	Flight Attendant Complaints	38
American	longlines	33

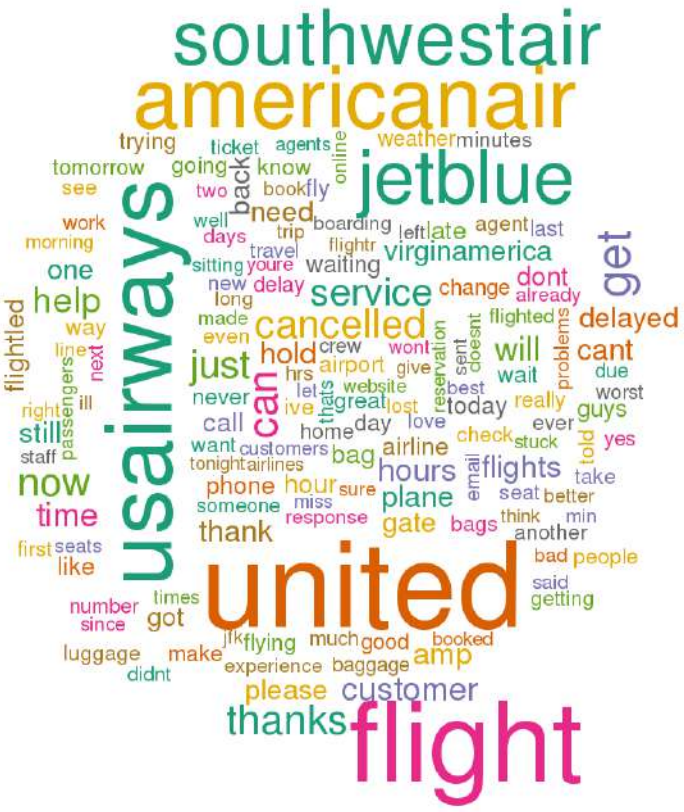
Southwest	longlines	29
Virgin America	Flight Booking Problems	28
United	Damaged Luggage	22
Virgin America	Can’t Tell	22
Virgin America	Bad Flight	19
Virgin America	Cancelled Flight	18
Virgin America	Late Flight	17
Delta	longlines	14
Southwest	Damaged Luggage	14
American	Damaged Luggage	11
Delta	Damaged Luggage	11
US Airways	Damaged Luggage	11
Virgin America	Flight Attendant Complaints	5

Virgin America	Lost Luggage	5
Virgin America	Damaged Luggage	4
Virgin America	longlines	3

Another question that comes to mind - what terms tend to appear in these tweets? We'll create a quick helper function to help create some wordclouds and then find out.

```
library(tm)

library

(wordcloud)

makeWordCloud <- function(documents) {

  corpus = Corpus(VectorSource(tolower(documents)))

  corpus = tm_map(corpus, removePunctuation)

  corpus = tm_map(corpus, removeWords, stopwords("english"))


  frequencies = DocumentTermMatrix(corpus)

  word_frequencies = as.data.frame(as.matrix(frequencies))


  words <- colnames(word_frequencies)

  freq <- colSums(word_frequencies)

  wordcloud(words, freq,

            min.freq=sort(freq, decreasing=TRUE)[[150]],

            colors=brewer.pal(8, "Dark2"),

            random.color=TRUE)

}
```

```
makeWordCloud(dbGetQuery(db, "SELECT text FROM Tweets")$text)
```



This was just intended to be a quick introduction to the dataset exploring it - I encourage you to dig deeper and I'm curious to see what you find!

Conclusion:

Employing NLP techniques is fundamental to creating a robust sentiment analysis solution. From data preprocessing to machine learning models and aspect-based analysis, NLP empowers you to extract valuable insights from textual data. By continually improving your approach and staying updated with the latest advancements, you can build a highly effective sentiment analysis solution that provides valuable insights into customer feedback and public sentiment.