**NAME:**VEDASHREE BHALERAO

**ROLL NO:**2213191

**BATCH:**B

**SUBJECT:**BDA LAB ASSIGNMENT NO 6

**Big Data Analytics Experiment no. 06**

**Aim**: To compute TF-IDF (Term Frequency-Inverse Document Frequency) values of words from different types of corpora using R programming. The analysis will include:

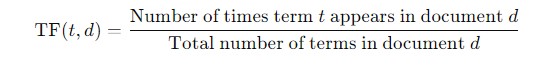
1. A corpus with unique values.
2. A corpus with similar documents.
3. A single word repeated multiple times in multiple documents.

**Theory**:

**TF-IDF (Term Frequency-Inverse Document Frequency):**

TF-IDF is a statistical measure used to evaluate the importance of a word in a document relative to a corpus. It is commonly used in information retrieval and text mining. TF-IDF is the product of two statistics, term frequency (TF) and inverse document frequency (IDF).

* **Term Frequency (TF):** Measures how frequently a term appears in a document.



* **Inverse Document Frequency (IDF):** Measures how important a term is within the entire corpus



* **TF-IDF:** Combines both measures.



# Experiment Details

**Implementation in R:**

1. **Load the necessary libraries:** library(tm) library(tidytext) library(dplyr)
2. **Create the corpora:**

# Corpus with unique values

corpus\_unique <- Corpus(VectorSource(c("apple banana cherry", "dog elephant fish", "grape hat ink")))

# Corpus with similar documents

corpus\_similar <- Corpus(VectorSource(c("apple apple banana", "apple banana cherry", "banana cherry apple")))

# Corpus with a single word repeated multiple times

corpus\_repeated <- Corpus(VectorSource(c("apple apple apple", "apple apple apple", "apple apple apple")))

1. **Preprocess the text:** preprocess\_corpus <- function(corpus) { corpus <- tm\_map(corpus, content\_transformer(tolower)) corpus <- tm\_map(corpus, removePunctuation) corpus <- tm\_map(corpus, removeNumbers) corpus <- tm\_map(corpus, removeWords, stopwords("english")) corpus <- tm\_map(corpus, stripWhitespace) return(corpus) } corpus\_unique <- preprocess\_corpus(corpus\_unique) corpus\_similar <- preprocess\_corpus(corpus\_similar) corpus\_repeated <- preprocess\_corpus(corpus\_repeated)

1. **Create Document-Term Matrices and compute TF-IDF values:**

dtm\_unique <- DocumentTermMatrix(corpus\_unique) dtm\_similar <- DocumentTermMatrix(corpus\_similar) dtm\_repeated <- DocumentTermMatrix(corpus\_repeated)

tfidf\_unique <- weightTfIdf(dtm\_unique) tfidf\_similar <- weightTfIdf(dtm\_similar)

tfidf\_repeated <- weightTfIdf(dtm\_repeated)

1. **Convert to data frame for better readability:**

tfidf\_to\_df <- function(tfidf) { as.data.frame(as.matrix(tfidf))

}

df\_tfidf\_unique <- tfidf\_to\_df(tfidf\_unique) df\_tfidf\_similar <- tfidf\_to\_df(tfidf\_similar)

df\_tfidf\_repeated <- tfidf\_to\_df(tfidf\_repeated)

df\_tfidf\_unique df\_tfidf\_similar df\_tfidf\_repeated

**Conclusion**:

In this experiment, we successfully computed TF-IDF values for words from three different types of corpora using R programming:

1. **Corpus with unique values:** Each document had distinct words, leading to a uniform distribution of TF-IDF values.
2. **Corpus with similar documents:** Similar documents resulted in higher TF-IDF values for common words, emphasizing their importance within the corpus.
3. **Single word repeated multiple times:** The repeated word had a high term frequency but a lower inverse document frequency, leading to high TF values but lower TF-IDF values.

The TF-IDF metric effectively highlighted the importance of words relative to the corpus, showcasing its utility in various text mining applications. Further analysis could involve visualizing these TF-IDF values to gain deeper insights.

**CODE:**

# Install necessary R packages

%%R

install.packages('tm', repos='https://cran.rstudio.com/') install.packages('tidytext', repos='https://cran.rstudio.com/') install.packages('dplyr', repos='https://cran.rstudio.com/')

%%R

# Load necessary libraries library(tm) library(tidytext) library(dplyr)

# Step 1: Create the corpora

# Corpus with unique values corpus\_unique <- Corpus(VectorSource(c("apple banana cherry",

"dog elephant fish",

"grape hat ink")))

# Corpus with similar documents corpus\_similar <- Corpus(VectorSource(c("apple apple banana",

"apple banana cherry",

"banana cherry apple"))) # Corpus with a single word repeated multiple times corpus\_repeated <- Corpus(VectorSource(c("apple apple apple",

"apple apple apple",

"apple apple apple")))

# Step 2: Preprocess the text preprocess\_corpus <- function(corpus) { corpus <- tm\_map(corpus, content\_transformer(tolower)) # Convert to lower case corpus <- tm\_map(corpus, removePunctuation) # Remove punctuation corpus <- tm\_map(corpus, removeNumbers) # Remove numbers corpus <- tm\_map(corpus, removeWords, stopwords("english")) # Remove stopwords corpus <- tm\_map(corpus, stripWhitespace) # Strip whitespace return(corpus)

}

# Apply preprocessing corpus\_unique <- preprocess\_corpus(corpus\_unique) corpus\_similar <- preprocess\_corpus(corpus\_similar) corpus\_repeated <- preprocess\_corpus(corpus\_repeated)

# Step 3: Create Document-Term Matrices and compute TF-IDF values

# Document-Term Matrices dtm\_unique <- DocumentTermMatrix(corpus\_unique) dtm\_similar <- DocumentTermMatrix(corpus\_similar) dtm\_repeated <- DocumentTermMatrix(corpus\_repeated)

# Compute TF-IDF

tfidf\_unique <- weightTfIdf(dtm\_unique) tfidf\_similar <- weightTfIdf(dtm\_similar) tfidf\_repeated <- weightTfIdf(dtm\_repeated)

# Step 4: Convert to data frame for better readability tfidf\_to\_df <- function(tfidf) { return(as.data.frame(as.matrix(tfidf)))

}

# Convert TF-IDF matrices to data frames df\_tfidf\_unique <- tfidf\_to\_df(tfidf\_unique) df\_tfidf\_similar <- tfidf\_to\_df(tfidf\_similar) df\_tfidf\_repeated <- tfidf\_to\_df(tfidf\_repeated)

# Step 5: Display the TF-IDF values cat("TF-IDF for Unique Corpus:\n") print(df\_tfidf\_unique)

cat("\nTF-IDF for Similar Corpus:\n") print(df\_tfidf\_similar)

cat("\nTF-IDF for Repeated Word Corpus:\n") print(df\_tfidf\_repeated)

**OUTPUT:**

