US Airline Twitter Sentiment Analysis

Jazmyn

Introduction

This report analyzes the **US Airline Twitter Sentiment dataset**.
It includes Exploratory Data Analysis (EDA), word-level sentiment, and machine learning models.

```r
library(tidyverse)
library(tidytext)
library(ggplot2)
library(wordcloud)
library(RColorBrewer)
library(caret)
library(tm)
library(e1071)

# Load dataset
airline <- read_csv("Tweets.csv")
glimpse(airline)
```

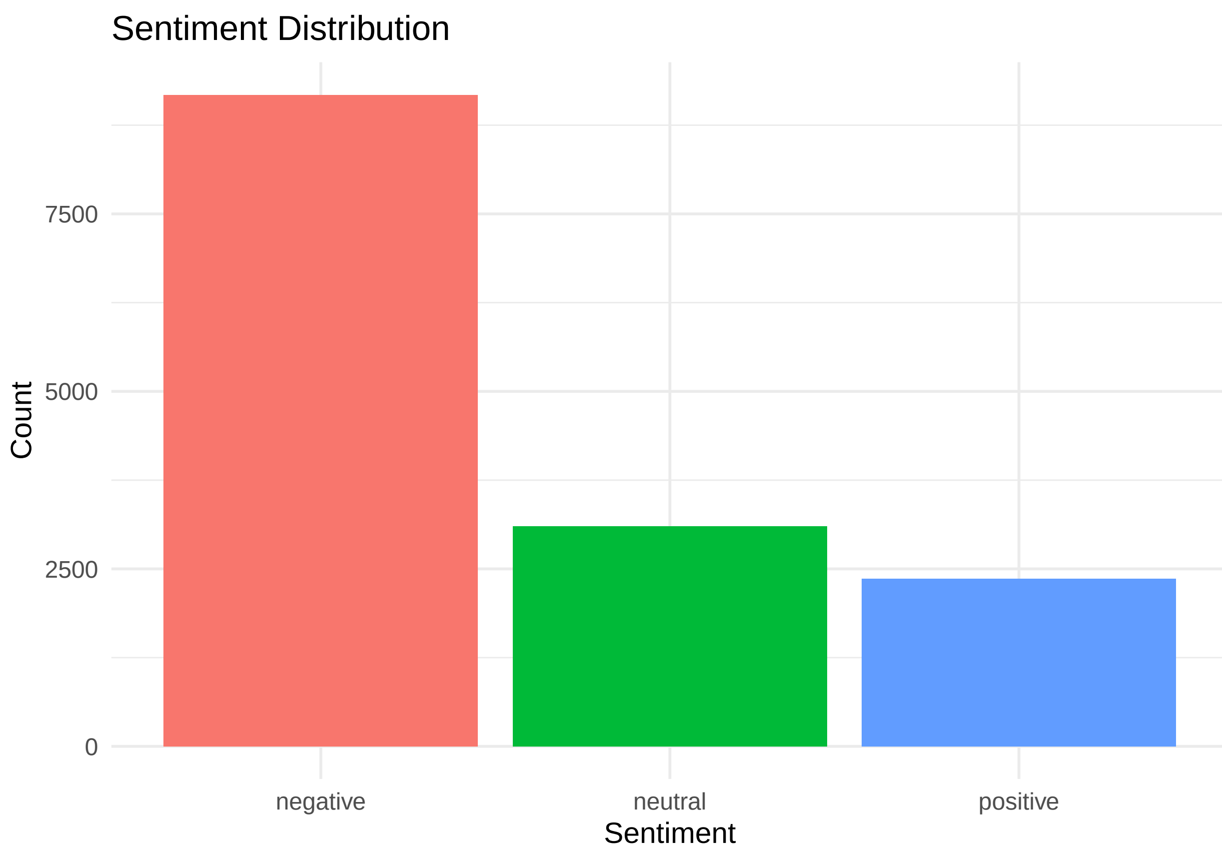
```
## Rows: 14,640
## Columns: 15
## $ tweet_id                     <dbl> 5.703061e+17, 5.703011e+17, 5.703011e+17,~
## $ airline_sentiment            <chr> "neutral", "positive", "neutral", "negati~
## $ airline_sentiment_confidence <dbl> 1.0000, 0.3486, 0.6837, 1.0000, 1.0000, 1~
## $ negativereason               <chr> NA, NA, NA, "Bad Flight", "Can't Tell", "~
## $ negativereason_confidence    <dbl> NA, 0.0000, NA, 0.7033, 1.0000, 0.6842, 0~
## $ airline                      <chr> "Virgin America", "Virgin America", "Virg~
## $ airline_sentiment_gold       <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ name                         <chr> "cairdin", "jnardino", "yvonnalynn", "jna~
## $ negativereason_gold          <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ retweet_count                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ text                         <chr> "@VirginAmerica What @dhepburn said.", "@~
## $ tweet_coord                  <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ tweet_created                <chr> "2015-02-24 11:35:52 -0800", "2015-02-24 ~
## $ tweet_location               <chr> NA, NA, "Lets Play", NA, NA, NA, "San Fra~
## $ user_timezone                <chr> "Eastern Time (US & Canada)", "Pacific Ti~
```

```r
summary(airline)
```

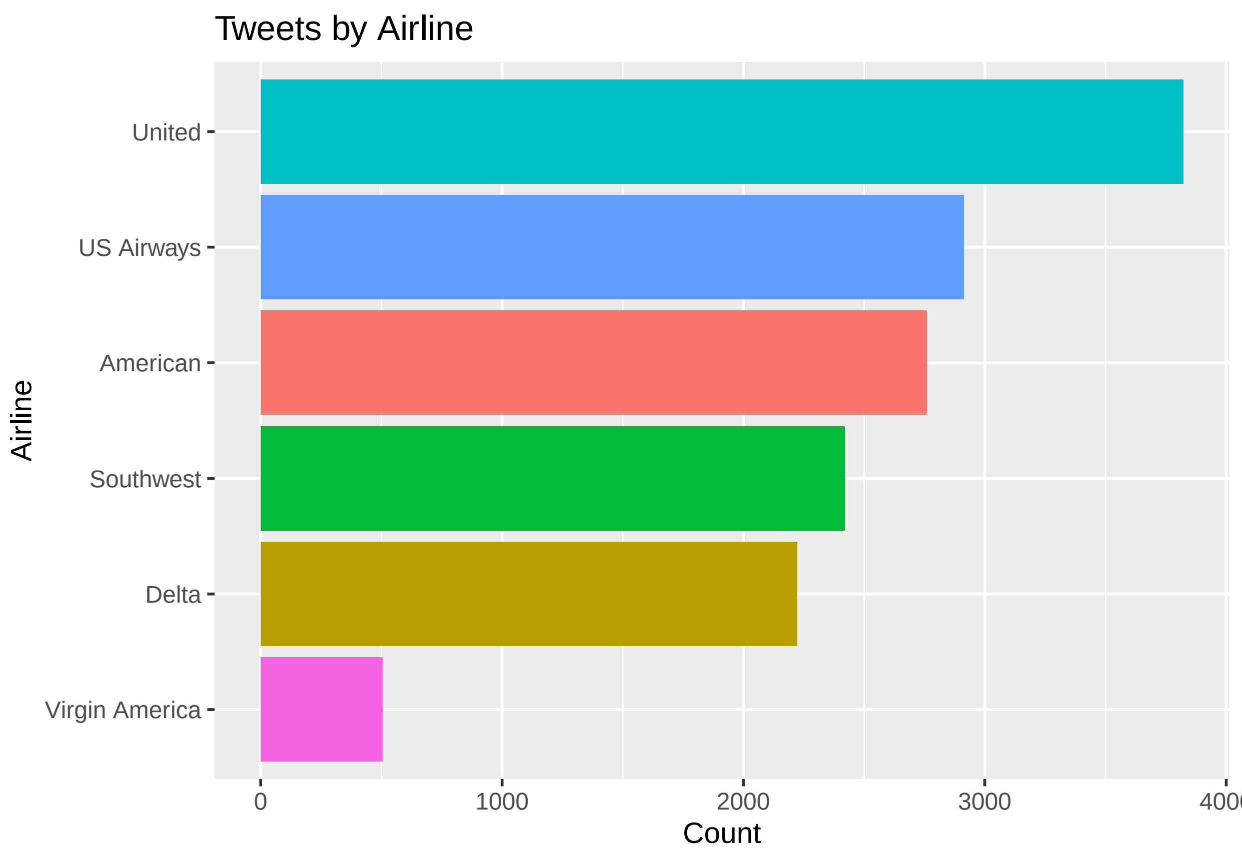
```
##     tweet_id          airline_sentiment  airline_sentiment_confidence
##  Min.   :5.676e+17   Length:14640       Min.   :0.3350
##  1st Qu.:5.686e+17   Class :character   1st Qu.:0.6923
##  Median :5.695e+17   Mode  :character   Median :1.0000
```

```
##  Mean   :5.692e+17                     Mean   :0.9002
##  3rd Qu.:5.699e+17                     3rd Qu.:1.0000
##  Max.   :5.703e+17                     Max.   :1.0000
##
##  negativereason     negativereason_confidence   airline
##  Length:14640       Min.   :0.0000            Length:14640
##  Class :character   1st Qu.:0.3606            Class :character
##  Mode  :character   Median :0.6706            Mode  :character
##                     Mean   :0.6383
##                     3rd Qu.:1.0000
##                     Max.   :1.0000
##                     NA's   :4118
##  airline_sentiment_gold     name           negativereason_gold
##  Length:14640           Length:14640       Length:14640
##  Class :character       Class :character   Class :character
##  Mode  :character       Mode  :character   Mode  :character
##
##
##
##
##  retweet_count         text           tweet_coord        tweet_created
##  Min.   : 0.00000   Length:14640       Length:14640       Length:14640
##  1st Qu.: 0.00000   Class :character   Class :character   Class :character
##  Median : 0.00000   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 0.08265
##  3rd Qu.: 0.00000
##  Max.   :44.00000
##
##  tweet_location     user_timezone
##  Length:14640       Length:14640
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
##
```

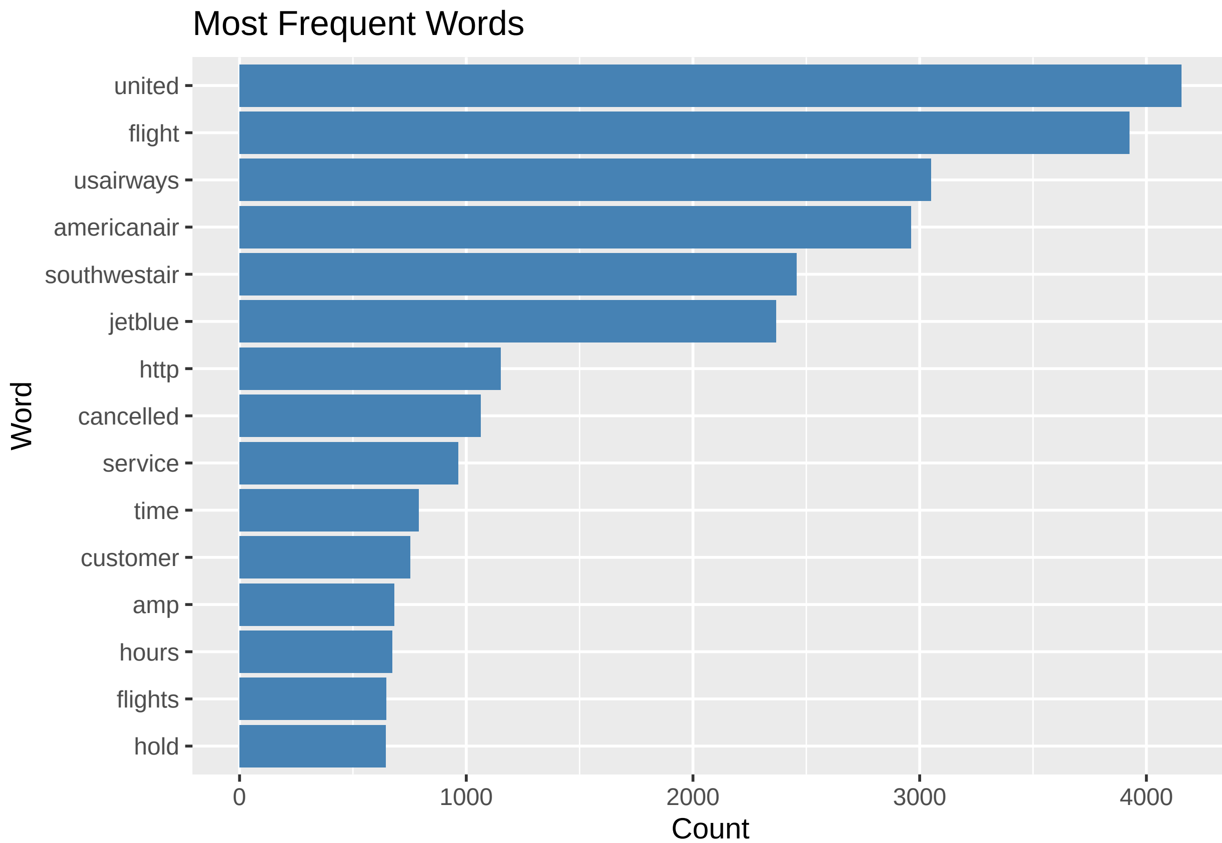
```r
# Sentiment distribution
sent_dist <- airline %>% count(airline_sentiment)
ggplot(sent_dist, aes(x = airline_sentiment, y = n, fill = airline_sentiment)) +
  geom_col(show.legend = FALSE) +
  labs(title = "Sentiment Distribution", x = "Sentiment", y = "Count") +
  theme_minimal()
```



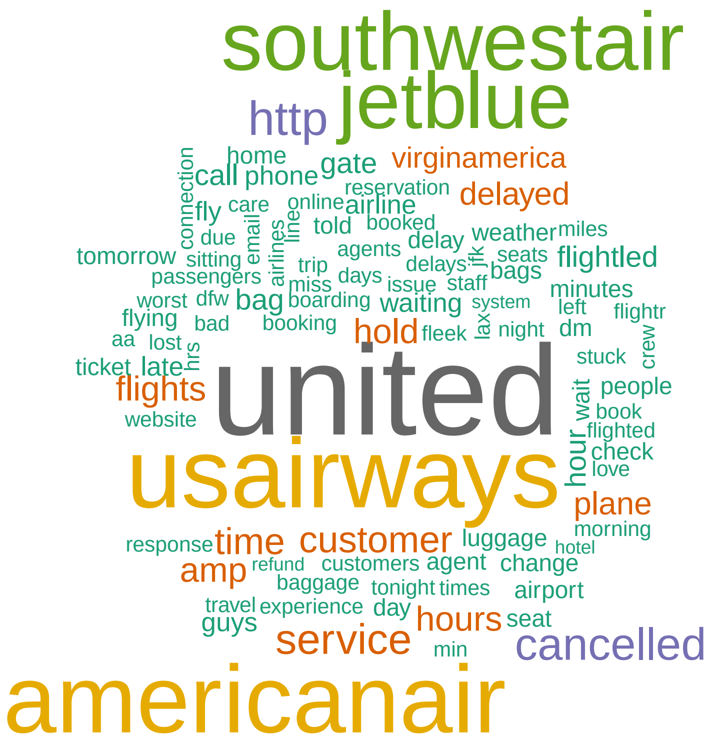

```r
# Tweets per airline
airline %>% count(airline) %>%
  ggplot(aes(x = reorder(airline, n), y = n, fill = airline)) +
  geom_col(show.legend = FALSE) +
  coord_flip() +
  labs(title = "Tweets by Airline", x = "Airline", y = "Count")
```



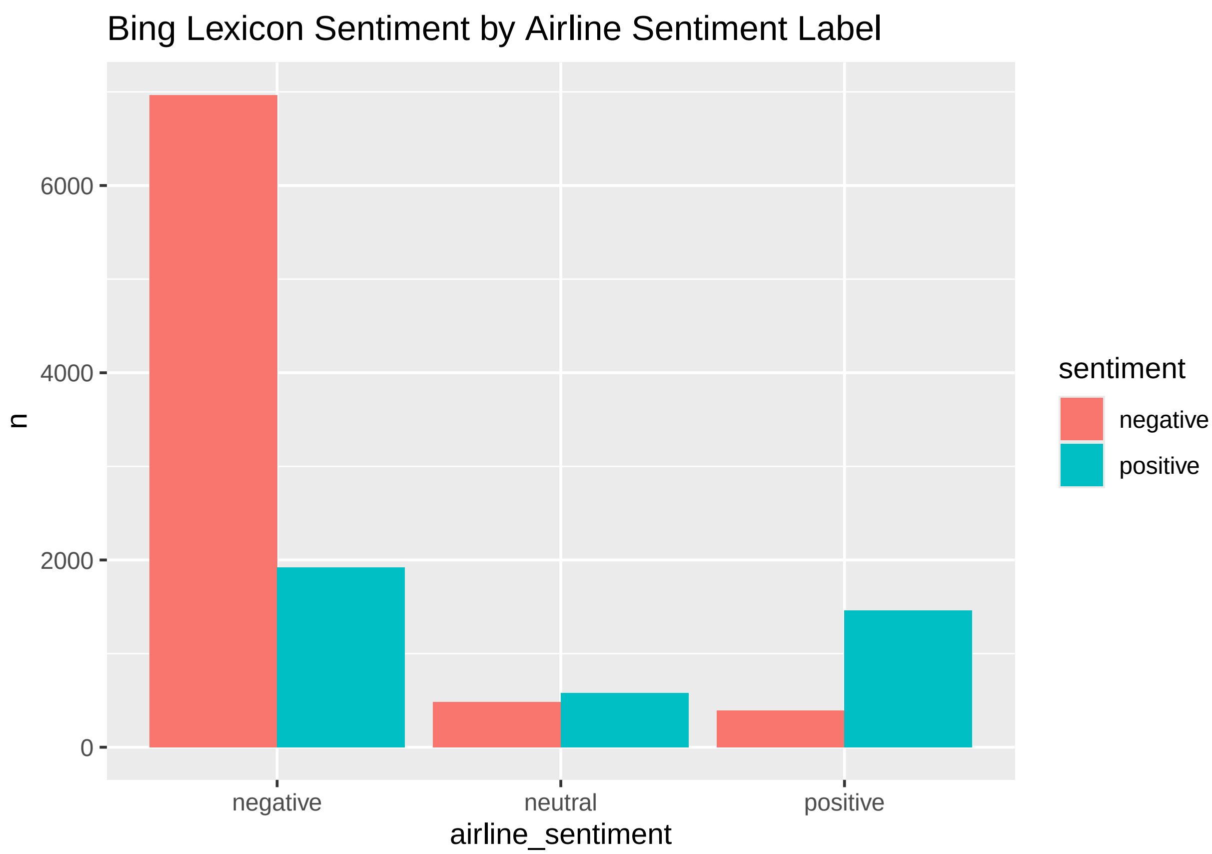

```r
# Tokenize and clean
tidy_tweets <- airline %>%
  select(airline, airline_sentiment, text) %>%
  unnest_tokens(word, text) %>%
  filter(!word %in% stop_words$word,
         str_detect(word, "^[a-z']+$"))

# Word frequency
word_freq <- tidy_tweets %>% count(word, sort = TRUE)
ggplot(head(word_freq, 15), aes(x = reorder(word, n), y = n)) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(title = "Most Frequent Words", x = "Word", y = "Count")
```



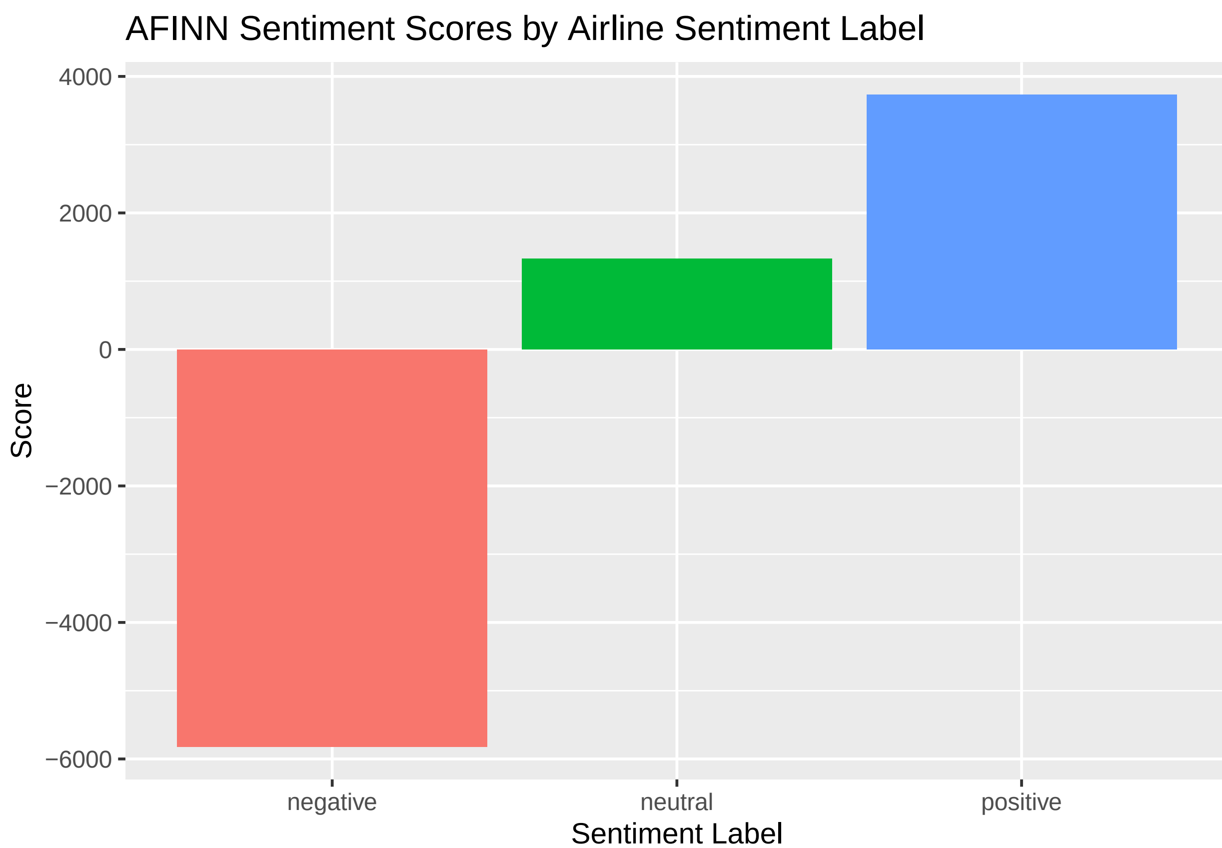

```r
# Wordcloud
set.seed(123)
wordcloud(words = word_freq$word, freq = word_freq$n,
          max.words = 100, colors = brewer.pal(8, "Dark2"))
```




```r
# Sentiment lexicon (Bing)
bing <- get_sentiments("bing")
bing_sent <- tidy_tweets %>%
  inner_join(bing, by = "word") %>%
  count(airline_sentiment, sentiment)
ggplot(bing_sent, aes(x = airline_sentiment, y = n, fill = sentiment)) +
  geom_col(position = "dodge") +
  labs(title = "Bing Lexicon Sentiment by Airline Sentiment Label")
```




```r
# AFINN sentiment
afinn <- get_sentiments("afinn")
afinn_sent <- tidy_tweets %>%
  inner_join(afinn, by = "word") %>%
  group_by(airline_sentiment) %>%
  summarise(score = sum(value), .groups = "drop")

ggplot(afinn_sent, aes(x = airline_sentiment, y = score, fill = airline_sentiment)) +
  geom_col(show.legend = FALSE) +
  labs(title = "AFINN Sentiment Scores by Airline Sentiment Label",
       x = "Sentiment Label", y = "Score")
```




```r
# Prepare for modeling
airline$text <- tolower(airline$text)
corpus <- VCorpus(VectorSource(airline$text))
corpus <- tm_map(corpus, content_transformer(tolower))
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, removeWords, stopwords("en"))
corpus <- tm_map(corpus, stripWhitespace)

dtm <- DocumentTermMatrix(corpus)
dtm <- removeSparseTerms(dtm, 0.995)

dataset <- as.data.frame(as.matrix(dtm))
dataset$sentiment <- as.factor(airline$airline_sentiment)

# Train-test split
set.seed(123)
trainIndex <- createDataPartition(dataset$sentiment, p = 0.7, list = FALSE)
train <- dataset[trainIndex, ]
test <- dataset[-trainIndex, ]
train$sentiment <- factor(train$sentiment, levels = levels(dataset$sentiment))
test$sentiment <- factor(test$sentiment, levels = levels(dataset$sentiment))

# Naive Bayes model
nb_model <- naiveBayes(sentiment ~ ., data = train)
nb_preds <- predict(nb_model, newdata = test)
```

```r
confusionMatrix(nb_preds, test$sentiment)
```

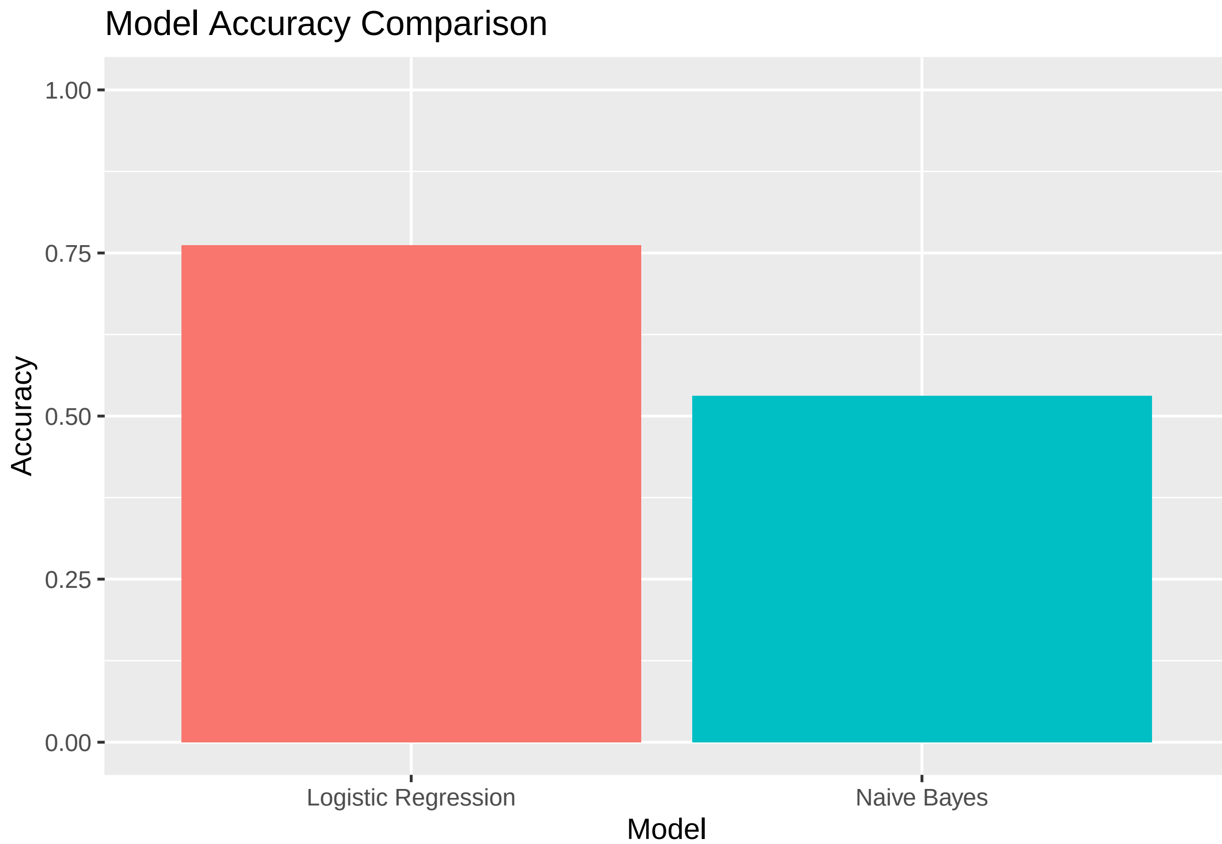
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction negative neutral positive
##   negative     1390      85       58
##   neutral       536     362       69
##   positive      827     482      581
##
## Overall Statistics
##
##                Accuracy : 0.5314
##                  95% CI : (0.5165, 0.5463)
##     No Information Rate : 0.6271
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2954
##
##  Mcnemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##                      Class: negative Class: neutral Class: positive
## Sensitivity                   0.5049        0.38967          0.8206
## Specificity                   0.9126        0.82520          0.6445
## Pos Pred Value                0.9067        0.37435          0.3074
## Neg Pred Value                0.5229        0.83436          0.9492
## Prevalence                    0.6271        0.21162          0.1613
## Detection Rate                0.3166        0.08246          0.1323
## Detection Prevalence          0.3492        0.22027          0.4305
## Balanced Accuracy             0.7088        0.60743          0.7326
```

```r
# Logistic regression (glmnet)
log_model <- train(sentiment ~ ., data = train,
                   method = "glmnet",
                   trControl = trainControl(method = "cv", number = 3))
log_preds <- predict(log_model, newdata = test)
confusionMatrix(log_preds, test$sentiment)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction negative neutral positive
##   negative     2466     392      189
##   neutral       199     455       95
##   positive       88      82      424
##
## Overall Statistics
##
##                Accuracy : 0.762
##                  95% CI : (0.7491, 0.7745)
```

```
##     No Information Rate : 0.6271
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5303
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: negative Class: neutral Class: positive
## Sensitivity                   0.8958         0.4898         0.59887
## Specificity                   0.6451         0.9151         0.95383
## Pos Pred Value                0.8093         0.6075         0.71380
## Neg Pred Value                0.7863         0.8698         0.92518
## Prevalence                    0.6271         0.2116         0.16128
## Detection Rate                0.5617         0.1036         0.09658
## Detection Prevalence          0.6941         0.1706         0.13531
## Balanced Accuracy             0.7704         0.7024         0.77635
```

```r
# Accuracy comparison
model_results <- tibble(
  Model = c("Naive Bayes", "Logistic Regression"),
  Accuracy = c(mean(nb_preds == test$sentiment),
               mean(log_preds == test$sentiment))
)

ggplot(model_results, aes(x = Model, y = Accuracy, fill = Model)) +
  geom_col(show.legend = FALSE) +
  ylim(0, 1) +
  labs(title = "Model Accuracy Comparison")
```


Model Accuracy Comparison
1.00
0.75
0.50
0.25
0.00
Accuracy
Logistic Regression
Naive Bayes
Model