Business Case Study

Promothesh Chatterjee*

Business Case Study: Enhancing Customer Experience through Sentiment Analysis of Reviews

Background **Company:** XYZ Electronics

Industry: E-commerce

Problem Statement: XYZ Electronics, a leading online retailer of consumer electronics, has been struggling with mixed customer reviews. While some customers rave about the products and services, others express significant dissatisfaction. This inconsistency in customer feedback poses a challenge for the company to identify and address specific areas of concern, ultimately impacting customer loyalty and brand reputation.

Detailed Business Problem

Current Situation XYZ Electronics has a diverse range of products, including smartphones, laptops, home appliances, and accessories. The company prides itself on offering high-quality products and excellent customer service. However, despite these efforts, customer reviews on their website and other platforms reveal a troubling pattern:

- **Positive Reviews:** Many customers express high satisfaction with the quality and performance of the products, timely delivery, and responsive customer service. Phrases such as "great product," "excellent service," and "highly recommend" are commonly found in positive reviews.
- **Negative Reviews:** A significant portion of customers report dissatisfaction, citing issues like product defects, poor customer service, delayed shipping, and misleading product descriptions. Phrases such as "worst experience," "poor quality," and "never buy again" are frequently mentioned in negative reviews.

Impact on Business The mixed nature of the reviews has several adverse effects on XYZ Electronics:

1. **Brand Reputation:** Negative reviews can tarnish the company's image, deterring potential customers from making purchases and eroding trust in the brand.
2. **Customer Retention:** Dissatisfied customers are less likely to make repeat purchases and more likely to switch to competitors, resulting in a loss of lifetime value.
3. **Operational Inefficiencies:** Without a systematic approach to categorizing and analyzing feedback, the company struggles to pinpoint specific issues, leading to inefficiencies in addressing customer concerns.
4. **Marketing and Sales:** Inconsistent reviews create challenges in marketing campaigns and sales strategies. Positive reviews can drive sales, but negative reviews can significantly hinder potential growth.

Strategic Objectives

To address these challenges, XYZ Electronics has set the following strategic objectives:

1. **Enhance Customer Satisfaction:** Identify and resolve the root causes of customer dissatisfaction to improve overall satisfaction and loyalty.
2. **Improve Product and Service Quality:** Use customer feedback to guide product development and service improvements, ensuring that offerings meet or exceed customer expectations.
3. **Strengthen Brand Reputation:** Build a positive brand image by consistently delivering high-quality products and services, and by effectively managing and responding to customer feedback.
4. **Boost Sales and Market Share:** Leverage positive customer feedback in marketing campaigns to attract new customers and increase market share, while mitigating the impact of negative reviews.

Proposed Strategy

XYZ Electronics plans to implement a comprehensive sentiment analysis solution to achieve these objectives. This solution will involve the following steps:

1. **Data Collection:** Gather and aggregate customer reviews from various sources, including the company website, social media, and third-party review platforms.
2. **Text Analysis:** Apply natural language processing (NLP) techniques to analyze the sentiment of customer reviews, categorizing them into positive, negative, and neutral sentiments.
3. **Actionable Insights:** Use the insights gained from sentiment analysis to inform business decisions, focusing on product improvements, service enhancements, and targeted marketing efforts.
4. **Continuous Improvement:** Establish a feedback loop to continuously monitor and analyze new reviews, ensuring that the company remains responsive to customer needs and can swiftly address any emerging issues.

Implementation and Expected Outcomes

By implementing this sentiment analysis strategy, XYZ Electronics expects to achieve the following outcomes:

- **Increased Customer Satisfaction:** By proactively addressing common issues and improving products and services based on customer feedback, the company aims to boost overall customer satisfaction.
- **Enhanced Brand Loyalty:** Satisfied customers are more likely to become repeat buyers and brand advocates, helping to build a loyal customer base.
- **Improved Product Quality:** Detailed insights into customer feedback will guide product development teams in enhancing product features and quality, leading to fewer defects and returns.

- **Stronger Brand Reputation:** A systematic approach to managing and responding to customer feedback will help build a positive brand image and attract new customers.

- **Higher Sales and Revenue:** Positive reviews and improved customer satisfaction will drive sales growth and increase market share, ultimately contributing to higher revenue.

Conclusion

XYZ Electronics recognizes the critical importance of understanding and addressing customer feedback. By leveraging advanced sentiment analysis techniques, the company can transform customer reviews into valuable insights, driving improvements in products, services, and overall customer experience. This strategic approach will not only enhance customer satisfaction and loyalty but also strengthen the company's competitive position in the market.

Detailed Explanation of the R Code

The provided R code is designed to address the business problem by building a predictive model to classify customer reviews into positive, negative, and neutral categories. Here's a step-by-step explanation of the code:

```r
library(text)
library(caret)
library(dplyr)
library(glmnet)
```

1. Import Necessary Libraries

- **text:** Used for generating embeddings from text data.
- **caret:** Provides functions for creating data partitions, training models, and evaluating performance.
- **dplyr:** Used for data manipulation.
- **glmnet:** Used for fitting generalized linear models via penalized maximum likelihood.

```r
set.seed(123)
n <- 200

positive_reviews <- c(
  "This product is great, I love it!", "Absolutely fantastic service.",
  "I am very satisfied with my purchase.", "Excellent quality, highly recommend.",
  "This product exceeded my expectations.", "Very happy with the quality.",
```

```r
  "The service was exceptional.", "I will definitely buy again.",
  "Highly recommended product.", "Great value for the price."
)

negative_reviews <- c(
  "Worst experience ever, very disappointed.", "Terrible quality, do not buy this.",
  "I hate this product, it's the worst.", "Very bad, not happy with the product.",
  "This product is a waste of money.", "Not satisfied with the purchase.",
  "The quality is horrible.", "I will never buy this again.",
  "Poor value for the price.", "The product broke after one use."
)

noisy_reviews <- c(
  "This product is okaay, not great.", "I am somewhat satisfied.",
  "The product is average, nothing special.", "Service was okay, could be better.",
  "The quality is decent, but not exceptional.", "The experience was just alright.",
  "It's a mediocre product.", "Not too bad, but not great either.",
  "The product is fine, nothing more.", "Average quality, expected better."
)
```

2. Create a Synthetic Dataset

- **positive_reviews:** List of simulated positive customer reviews.
- **negative_reviews:** List of simulated negative customer reviews.
- **noisy_reviews:** List of simulated neutral or noisy customer reviews that do not clearly indicate positive or negative sentiment.

```r
text_data <- data.frame(
  review = sample(c(positive_reviews, negative_reviews, noisy_reviews), n, replace = TRUE),
  stringsAsFactors = FALSE
)
```

3. Generate Random Reviews

- **text_data:** Data frame containing randomly sampled reviews from the three lists created above.

```r
text_data$label <- ifelse(grepl("great|love|fantastic|satisfied|excellent|highly|exceeded|happy|excepti
text_data$label <- ifelse(grepl("worst|terrible|hate|bad|waste|not satisfied|horrible|never|poor|broke"
text_data$label <- ifelse(grepl("okay|somewhat|average|mediocre|decent|alright|fine", text_data$review)
```

4. **Assign Labels to Reviews**

- **label:** Binary labels assigned based on the presence of specific keywords in the reviews. Positive reviews are labeled as `1`, negative reviews as `0`, and neutral/noisy reviews are assigned a random label to introduce variability.

```
table(text_data$label)
```

5. **Check Class Distribution**

- **table:** Displays the distribution of positive and negative labels in the dataset.

```
embeddings <- textEmbed(
  texts = text_data$review,
  model = 'bert-base-uncased',
  layers = 11:12,
  aggregation_from_layers_to_tokens = "mean",
  aggregation_from_tokens_to_texts = "mean"
)
```

6. **Generate Embeddings for Text Data**

Detailed Explanation of Step 6: Generating Embeddings for Text Data

In step 6, the R code generates embeddings for the text data using the `textEmbed` function from the `text` package. This step is crucial for transforming the raw text reviews into a format that can be used for machine learning models. Here's a detailed breakdown of the process:

Purpose of Text Embeddings Text embeddings are dense vector representations of text data. They capture the semantic meaning of the text and enable machine learning models to process and learn from textual information. By using embeddings, we can convert text data into numerical format, making it suitable for input into models like Lasso regression.

Function Used: `textEmbed` The `textEmbed` function from the `text` package is used to generate embeddings for the text data. It utilizes pre-trained models from Hugging Face's Transformers library.

```
embeddings <- textEmbed(
  texts = text_data$review,
  model = 'bert-base-uncased',
  layers = 11:12,
  aggregation_from_layers_to_tokens = "mean",
  aggregation_from_tokens_to_texts = "mean"
)
```

Parameters of `textEmbed`

1. **texts = text_data$review:**
   - This parameter specifies the text data for which embeddings need to be generated. In this case, it is the `review` column of the `text_data` data frame, which contains customer reviews.
2. **model = 'bert-base-uncased':**
   - This parameter indicates the pre-trained model to be used for generating embeddings. The model 'bert-base-uncased' is a popular model from Hugging Face's library. BERT (Bidirectional Encoder Representations from Transformers) is known for its strong performance in various NLP tasks.
3. **layers = 11:12:**
   - BERT models consist of multiple layers. This parameter specifies which layers to use for generating embeddings. Here, the last two layers (11 and 12) of the BERT model are used. The final layers typically capture higher-level semantic information.
4. **aggregation_from_layers_to_tokens = "mean":**
   - This parameter defines how to aggregate embeddings across the specified layers for each token in the text. Using the mean aggregation, embeddings from the 11th and 12th layers are averaged for each token.
5. **aggregation_from_tokens_to_texts = "mean":**
   - This parameter defines how to aggregate token embeddings to represent the entire text. By averaging token embeddings, we obtain a single embedding vector for each review.

Process of Generating Embeddings

1. **Tokenization:**
   - The text data is first tokenized, meaning it is split into individual tokens (words or subwords).
2. **Embedding Extraction:**
   - The BERT model processes the tokens through its layers, generating embeddings at each layer. For this task, embeddings from the 11th and 12th layers are used.
3. **Aggregation:**

- Embeddings from the specified layers are averaged to create a single embedding for each token.
- The token embeddings are then averaged to produce a single embedding vector representing the entire review.

Result of `textEmbed` The `textEmbed` function returns an object containing the embeddings for the text data. These embeddings are stored in the `texts` component of the returned object and are added to the original `text_data` data frame.

```
text_data <- cbind(text_data, embeddings$texts)
```

- **cbind:** This function combines the original `text_data` data frame with the generated embeddings. Each review in the `text_data` data frame now has a corresponding embedding vector, making it suitable for input into the machine learning model.

Importance of Text Embeddings

Using text embeddings is essential because they capture the semantic meaning of the reviews in a numerical format. This allows the subsequent machine learning model to understand and learn from the textual data, improving its ability to classify reviews accurately.

By generating high-quality embeddings with a powerful model like BERT, the code ensures that the predictive model has rich and meaningful input features, leading to better performance in classifying customer reviews.

- **cbind:** Combines the generated embeddings with the original text data, creating a comprehensive dataset with both the original reviews and their corresponding embeddings.

```
set.seed(123)
trainIndex <- createDataPartition(text_data$label, p = 0.8, list = FALSE)
trainData <- text_data[trainIndex,]
testData <- text_data[-trainIndex,]
```

8. Split Data into Training and Test Sets

- **createDataPartition:** Splits the dataset into training (80%) and test (20%) sets.

```
trainData$label <- as.factor(trainData$label)
testData$label <- as.factor(testData$label)
```

9. Ensure Labels are Factors

- **as.factor:** Converts the label column to factors, which is required for classification tasks.

```r
nzv <- nearZeroVar(trainData, saveMetrics = TRUE)
trainData <- trainData[, !nzv$nzv]
testData <- testData[, !nzv$nzv]
```

10. Remove Near-Zero Variance Predictors

- **nearZeroVar:** Identifies predictors with near-zero variance, which are removed from the training and test datasets to improve model performance.

```r
x_train <- as.matrix(trainData[, -which(names(trainData) == "label" | names(trainData) == "review")])
y_train <- trainData$label

x_test <- as.matrix(testData[, -which(names(testData) == "label" | names(testData) == "review")])
y_test <- testData$label
```

11. Prepare Data for Modeling

- **as.matrix:** Converts the training and test data into matrix format, excluding the label and review columns, which are not needed for modeling.

```r
cv_model <- cv.glmnet(x_train, y_train, family = "binomial", alpha = 1, nfolds = 10)
```

12. Train Lasso Regression Model with Cross-Validation

- **cv.glmnet:** Trains a Lasso regression model using cross-validation (10 folds) to identify the best lambda parameter for regularization.

```r
predictions <- predict(cv_model, s = "lambda.min", newx = x_test, type = "class")
predictions <- as.factor(predictions)
```

13. Make Predictions

- **predict:** Uses the trained model to make predictions on the test data, applying the optimal lambda parameter identified during cross-validation.
- **as.factor:** Converts the predictions to factors for evaluation.

```r
conf_matrix <- confusionMatrix(predictions, y_test)
print(conf_matrix)
```

14. Evaluate Model Performance

- **confusionMatrix:** Generates a confusion matrix to evaluate the model's performance, displaying metrics such as accuracy, precision, recall, and F1-score.

Conclusion

This R code effectively demonstrates how to build a predictive model to classify customer reviews into positive, negative, and neutral categories. By leveraging dynamic embeddings from a pre-trained BERT model and using Lasso regression for classification, the code provides a robust solution for sentiment analysis of customer reviews, addressing the business problem faced by XYZ Electronics.