Product Review Sentiment Analysis

AI-Powered Customer Feedback Intelligence System

Technical Documentation

Authors:
SONU - SVNA PRO TEAM

Contents

Abstract

This document presents a comprehensive overview of the Product Review Sentiment Analysis system, an intelligent web application designed to automate the extraction and analysis of customer feedback from PDF documents. The system leverages Natural Language Processing (NLP) techniques to classify sentiments and generate actionable insights through interactive visualizations. Built using modern Python libraries including Streamlit, TextBlob, and Plotly, the application addresses the critical challenge of processing large volumes of customer reviews efficiently while maintaining objective and consistent sentiment classification. The system achieves real-time sentiment analysis with polarity scoring, providing businesses with quantifiable metrics and visual analytics to support data-driven decision-making in customer experience management.

Chapter 1

Introduction

1.1 Background

In the contemporary digital marketplace, customer reviews represent a crucial source of business intelligence. Organizations across industries—from e-commerce platforms to hospitality services—generate thousands of customer reviews daily. These reviews contain invaluable insights regarding product quality, service satisfaction, and brand perception. However, the manual analysis of such large-scale feedback presents significant operational challenges, including time consumption, subjective interpretation, and inconsistent classification methodologies.

1.2 Motivation

The primary motivation behind developing this sentiment analysis system stems from the following industry-wide challenges:

- **Volume Overload**: Businesses regularly receive hundreds to thousands of reviews that require systematic analysis
- **Manual Analysis Limitations**: Human-driven review analysis is time-intensive and susceptible to subjective bias
- **Inconsistent Classification**: Manual sentiment detection lacks standardization, leading to unreliable insights
- **Pattern Recognition Difficulties**: Identifying trends and actionable patterns across large datasets proves challenging without automation
- **Quantification Issues**: Converting qualitative feedback into quantifiable metrics for strategic decision-making remains problematic

1.3 Objectives

The Product Review Sentiment Analysis system aims to achieve the following objectives:

1. Automate the extraction of textual content from PDF documents containing customer reviews
2. Implement robust sentiment classification using Natural Language Processing algorithms

3. Calculate precise polarity scores to quantify sentiment intensity
4. Generate interactive data visualizations for intuitive insight comprehension
5. Provide a real-time analytics dashboard with filtering and exploration capabilities
6. Enable data export functionality for further analysis and reporting

1.4 Scope

This system is designed to serve multiple stakeholders across various business functions:

- **E-commerce Businesses**: Scale product review analysis across extensive catalogs
- **Product Managers**: Monitor customer satisfaction trends and product performance metrics
- **Marketing Teams**: Identify brand sentiment patterns and customer pain points
- **Customer Service Teams**: Prioritize responses to negative feedback efficiently
- **Quality Assurance**: Detect product defects and quality issues through early warning signals
- **Market Research**: Conduct competitive sentiment analysis across product categories
- **Hospitality Industry**: Analyze guest feedback for restaurants, hotels, and service providers
- **Application Developers**: Track user review sentiment for mobile and web applications
- **Sales Teams**: Generate customer satisfaction metrics for performance evaluation
- **Business Analytics**: Create sentiment reports for stakeholder presentations

Chapter 2

Problem Statement

2.1 Current Challenges in Review Analysis

Organizations face significant obstacles in managing customer feedback effectively:

Volume Management: The exponential growth of digital reviews creates an overwhelming quantity of data that exceeds human processing capacity. Manual reading and categorization of reviews becomes impractical beyond a certain threshold.

Time Efficiency: Manual sentiment analysis requires substantial human resources and time investment, delaying the extraction of actionable insights and potentially missing time-sensitive issues.

Subjectivity and Bias: Human analysts introduce personal biases and inconsistencies in sentiment interpretation, leading to unreliable classification across different reviewers or time periods.

Scalability Issues: Traditional manual approaches cannot scale proportionally with business growth, creating bottlenecks in customer feedback analysis.

Quantification Challenges: Converting qualitative sentiments into quantifiable metrics for reporting and trend analysis proves difficult without standardized methodologies.

2.2 Solution Requirements

An effective solution must address the following requirements:

- Process bulk reviews in seconds rather than hours or days
- Provide objective, consistent sentiment classification free from human bias
- Generate quantifiable metrics (polarity scores) for measurable analysis
- Present insights through intuitive visual representations
- Support interactive data exploration and filtering
- Enable integration with existing business intelligence workflows

Chapter 3

System Architecture

3.1 Architectural Overview

The Product Review Sentiment Analysis system implements a four-layer architecture designed for modularity, scalability, and maintainability:

Layer 1: Input Layer

- Handles PDF file upload through web interface
- Validates file format and integrity
- Manages file processing workflow

Layer 2: Extraction & Processing Layer

- Extracts textual content from PDF documents using PyPDF2
- Segments continuous text into individual reviews using regex pattern matching
- Performs text cleaning and preprocessing

Layer 3: Analysis Layer

- Classifies sentiment using TextBlob NLP algorithms
- Calculates polarity scores for each review
- Aggregates statistical metrics across the dataset

Layer 4: Presentation Layer

- Displays comprehensive metrics dashboard
- Renders interactive charts and visualizations
- Provides sortable and filterable data tables
- Enables CSV export functionality

3.2 Data Flow

The system processes data through the following sequential pipeline:

1. **PDF Upload** → User uploads document via web interface
2. **Text Extraction** → PyPDF2 extracts raw text from all pages
3. **Review Segmentation** → Regex splits continuous text into discrete reviews
4. **Sentiment Analysis** → TextBlob analyzes each review independently
5. **Data Structuring** → Results organized into Pandas DataFrame
6. **Statistical Aggregation** → Metrics calculated across dataset
7. **Visualization Generation** → Plotly creates interactive charts
8. **Dashboard Rendering** → Streamlit displays complete analytics interface

Chapter 4

Technology Stack

4.1 Core Technologies

4.1.1 Python (v3.7+)

Primary programming language providing the foundational runtime environment. Selected for its extensive data science ecosystem and library support.

4.1.2 Streamlit

Purpose: Web application framework for rapid dashboard development

Advantages:

- Eliminates need for separate frontend development
- Instant dashboard creation with minimal code
- Built-in responsive layout system
- Real-time data updates and interactivity
- Native widget support for file uploads, filters, and controls

Key Components Utilized:

- `st.file_uploader()`: PDF document upload interface
- `st.columns()`: Multi-column responsive layouts (2-3 column configurations)
- `st.metric()`: KPI display with percentage indicators
- `st.plotly_chart()`: Interactive visualization rendering
- `st.dataframe()`: Sortable, filterable data tables
- `st.download_button()`: CSV export functionality
- `st.divider()`: Visual section separation
- `st.multiselect()`: Multi-option filter controls

4.1.3 TextBlob

Purpose: Natural Language Processing library for sentiment analysis
Technical Specifications:

- Built on NLTK (Natural Language Toolkit) and Pattern libraries
- Provides simplified API for common NLP tasks
- Implements pre-trained sentiment analysis models
- Requires no custom training or model configuration

Capabilities:

- Sentiment polarity and subjectivity scoring
- Part-of-speech tagging
- Text tokenization
- Noun phrase extraction
- Language translation support

Sentiment Analysis Methodology: TextBlob employs a lexicon-based approach using a pre-trained sentiment lexicon. Each word carries an associated polarity value, and the algorithm aggregates these values while considering modifiers, negations, and intensifiers to compute an overall polarity score.

4.1.4 PyPDF2

Purpose: PDF text extraction toolkit
Technical Specifications:

- Pure Python implementation (no external dependencies)
- Supports various PDF format specifications
- Page-by-page content processing

Key Classes and Methods:

- `PdfReader`: Primary class for PDF file handling
- `.pages`: Iterator for accessing individual pages
- `.extract_text()`: Method for extracting textual content from pages

4.1.5 Pandas

Purpose: Data manipulation and statistical analysis framework

Functionality in System:

- Structures review data in DataFrame format for efficient processing
- Enables vectorized operations for performance optimization
- Facilitates data filtering, sorting, and grouping
- Supports CSV export for interoperability

Statistical Operations:

- `value_counts()`: Frequency distribution by sentiment category
- `mean()`: Average polarity score calculation
- `mode()`: Most frequently occurring sentiment identification

4.1.6 Plotly

Purpose: Interactive data visualization library

Technical Architecture:

- JavaScript-based rendering engine
- Publication-quality graphics output
- Two-tiered API: Express (high-level) and Graph Objects (low-level)

Visualizations Implemented:

1. **Pie Chart** (`px.pie`): Sentiment distribution proportions
2. **Bar Chart** (`px.bar`): Categorical sentiment counts
3. **Histogram** (`px.histogram`): Polarity score distribution with 30 bins

4.1.7 Regular Expressions (re)

Purpose: Pattern-based text processing

Application: Review segmentation using pattern `r'\n\s*\n'` to split continuous text on paragraph breaks (double newlines).

4.1.8 Collections (Counter)

Purpose: Frequency counting and statistical analysis

Application: Efficiently counts occurrences of sentiment categories for aggregation.

Chapter 5

Sentiment Analysis Methodology

5.1 Polarity Score System

5.1.1 Score Range and Interpretation

The system employs a continuous polarity scale ranging from -1.0 to $+1.0$:

Positive Sentiment Range:

- $+0.8$ **to** $+1.0$: Extremely Positive (exceptional satisfaction, strong enthusiasm)
- $+0.5$ **to** $+0.8$: Strongly Positive (clear satisfaction, positive endorsement)
- $+0.1$ **to** $+0.5$: Moderately Positive (generally favorable, mild approval)

Neutral Sentiment Range:

- -0.1 **to** $+0.1$: Neutral (balanced feedback, factual statements, no strong emotion)

Negative Sentiment Range:

- -0.5 **to** -0.1: Moderately Negative (mild dissatisfaction, constructive criticism)
- -0.8 **to** -0.5: Strongly Negative (clear dissatisfaction, significant complaints)
- -1.0 **to** -0.8: Extremely Negative (severe dissatisfaction, strong criticism)

5.1.2 Classification Thresholds

The system implements threshold-based classification with a neutral buffer zone:

```python
if polarity > 0.1:
    sentiment = 'Positive'
elif polarity < -0.1:
    sentiment = 'Negative'
else:
    sentiment = 'Neutral'
```

Listing 5.1: Sentiment Classification Logic

Rationale for 0.1 Threshold:

- Creates a neutral buffer zone to reduce classification noise
- Prevents marginal sentiments from being misclassified as positive/negative
- Improves overall classification accuracy by acknowledging ambiguous sentiments
- Reduces false positive and false negative rates

5.2 Review Segmentation Algorithm

5.2.1 Challenge

PDF documents typically contain multiple reviews in continuous text format without explicit delimiters, requiring intelligent segmentation to isolate individual reviews.

5.2.2 Solution Implementation

Regex Pattern: `r’\n\s*\n’`

Pattern Components:

- `\n`: First newline character
- `\s*`: Zero or more whitespace characters (spaces, tabs)
- `\n`: Second newline character

Logic: Splits text on double newline sequences (paragraph breaks), which typically separate distinct reviews.

Post-Processing:

- Strips leading/trailing whitespace from each segment
- Filters out segments with less than 20 characters (removes noise)
- Returns list of valid review strings

```
reviews = re.split(r’\n\s*\n’, text)
reviews = [r.strip() for r in reviews if r.strip() and len(r.
    strip()) > 20]
```

Listing 5.2: Review Segmentation Implementation

5.3 Sentiment Classification Process

For each extracted review, the system:

1. Creates TextBlob object from review text
2. Extracts polarity score using `.polarity` attribute
3. Applies threshold-based classification logic
4. Returns sentiment category and numerical polarity score
5. Stores results in structured DataFrame

Chapter 6

System Features and Components

6.1 Dashboard Interface Components

6.1.1 Metrics Row (Three-Column Layout)

Displays high-level KPIs with percentage calculations:

- **Positive Reviews**: Count and percentage of total
- **Negative Reviews**: Count and percentage of total
- **Neutral Reviews**: Count and percentage of total

Each metric includes visual emoji indicators (, ,) for quick recognition.

6.1.2 Visualization Row (Two-Column Layout)

Column 1 - Pie Chart:

- Displays proportional sentiment distribution
- Color-coded segments: Green (Positive), Red (Negative), Blue (Neutral)
- Interactive tooltips with exact percentages

Column 2 - Bar Chart:

- Shows absolute sentiment counts
- Consistent color scheme with pie chart
- Hover functionality for precise values

6.1.3 Statistical Metrics Row (Two-Column Layout)

- **Average Polarity Score**: Mean polarity across all reviews (3 decimal precision)
- **Most Common Sentiment**: Mode of sentiment distribution

6.1.4 Distribution Analysis

Histogram Visualization:

- Displays polarity score distribution across 30 bins
- Reveals distribution shape (normal, skewed, bimodal)
- Identifies clustering patterns in sentiment intensity

6.1.5 Data Explorer

Interactive Features:

- Multi-select filter by sentiment categories
- Sortable columns (Review ID, Review text, Sentiment, Polarity)
- Pagination for large datasets (400px height viewport)
- Displays truncated reviews (200 characters) with ellipsis

6.1.6 Individual Review Analysis

- Dropdown selector for review ID navigation
- Full review text display (untruncated)
- Sentiment category and polarity score presentation
- Formatted info box for readability

6.1.7 Export Functionality

- CSV download button for complete dataset
- Includes all columns: Review_ID, Review text, Sentiment, Polarity_Score
- Facilitates integration with external analytics tools

Chapter 7

Implementation Details

7.1 Core Functions

7.1.1 `extract_text_from_pdf(pdf_file)`

Purpose: Extracts all textual content from uploaded PDF document
Parameters:

- `pdf_file`: File object from Streamlit uploader

Process:

1. Initializes PyPDF2.PdfReader object
2. Iterates through all pages using `.pages` iterator
3. Extracts text from each page using `.extract_text()`
4. Concatenates page text into single string

Returns: Complete extracted text as string

7.1.2 `get_sentiment(text)`

Purpose: Analyzes sentiment of individual review text
Parameters:

- `text`: String containing review content

Process:

1. Creates TextBlob object from input text
2. Extracts polarity score using `.polarity` attribute
3. Applies classification logic based on threshold values
4. Determines sentiment category

Returns: Tuple (sentiment_category, polarity_score)

7.1.3 `split_reviews(text)`

Purpose: Segments continuous text into individual reviews

Parameters:

- `text`: Complete extracted text from PDF

Process:

1. Applies regex split pattern to identify review boundaries
2. Strips whitespace from each segment
3. Filters segments by minimum length (20 characters)
4. Removes empty or invalid segments

Returns: List of review strings

7.2 Data Structure

The system organizes results in a Pandas DataFrame with the following schema:

Column	**Data Type**	**Description**
Review_ID	Integer	Sequential identifier (1 to N)
Review	String	Truncated review text (200 chars) for display
Full_Review	String	Complete untruncated review text
Sentiment	String	Categorical classification (Positive/Negative/Neutral)
Polarity_Score	Float	Numerical polarity value (3 decimal precision)

Table 7.1: DataFrame Schema

Chapter 8

User Workflow

8.1 Standard Operating Procedure

1. **Access Application**: Navigate to deployed Streamlit application URL
2. **Upload PDF**: Click "Upload PDF with Product Reviews" button and select file
3. **Automatic Processing**: System extracts, segments, and analyzes reviews automatically
4. **Review Metrics**: Examine high-level KPIs in metrics row
5. **Analyze Visualizations**: Interact with pie chart, bar chart, and histogram
6. **Explore Data**: Use filters to focus on specific sentiment categories
7. **Examine Individual Reviews**: Select review IDs to view complete text
8. **Export Results**: Download CSV file for external analysis or reporting

8.2 Instructions Display (No File Uploaded)

When no PDF is uploaded, the system displays instructional content:

- Clear usage instructions (numbered steps)
- Feature list highlighting key capabilities
- Upload prompt with visual indicator

Chapter 9

Technical Requirements

9.1 Dependencies

```
streamlit>=1.20.0
pandas>=1.5.0
PyPDF2>=3.0.0
textblob>=0.17.0
plotly>=5.13.0
```

Listing 9.1: Python Package Requirements

9.2 System Requirements

- **Python Version**: 3.7 or higher
- **Memory**: Minimum 2GB RAM (4GB recommended for large PDFs)
- **Storage**: Negligible (application processes files in-memory)
- **Browser**: Modern web browser (Chrome, Firefox, Safari, Edge)

9.3 Installation

```
pip install streamlit pandas PyPDF2 textblob plotly
```

Listing 9.2: Installation Command

9.4 Deployment

```
streamlit run app.py
```

Listing 9.3: Application Deployment

Application accessible at `http://localhost:8501` by default.

Chapter 10

Advantages and Benefits

10.1 Operational Efficiency

- **Speed**: Processes hundreds of reviews in seconds versus hours manually
- **Scalability**: Handles growing review volumes without proportional resource increase
- **Consistency**: Eliminates human bias and subjective interpretation variance

10.2 Business Intelligence

- **Quantifiable Metrics**: Converts qualitative feedback into measurable KPIs
- **Trend Identification**: Reveals patterns across large datasets
- **Prioritization**: Enables focus on critical negative feedback requiring immediate attention

10.3 Cost Effectiveness

- **Resource Optimization**: Reduces manual labor requirements for review analysis
- **Open Source**: Built entirely on free, open-source libraries
- **No Training Required**: Pre-trained models eliminate ML training overhead

10.4 User Experience

- **Intuitive Interface**: No technical expertise required for operation
- **Interactive Visualizations**: Engages users with dynamic, explorable charts
- **Flexible Export**: Integrates seamlessly with existing business workflows

Chapter 11

Limitations and Future Enhancements

11.1 Current Limitations

- **Language Support**: Currently optimized for English text only
- **PDF Format Dependency**: Requires text-based PDFs (not image-based scans)
- **Segmentation Assumptions**: Assumes paragraph breaks separate reviews
- **Context Understanding**: Lexicon-based approach may miss nuanced sarcasm or context-dependent sentiment

11.2 Proposed Enhancements

- **Multi-language Support**: Integration with translation APIs for global review analysis
- **OCR Integration**: Support for scanned PDF documents using Tesseract
- **Advanced NLP Models**: Integration with transformer-based models (BERT, RoBERTa) for improved accuracy
- **Topic Modeling**: Identify common themes and topics within sentiment categories
- **Time Series Analysis**: Track sentiment trends over time with temporal data
- **Comparative Analysis**: Multi-product sentiment comparison capabilities
- **Custom Thresholds**: User-configurable classification thresholds
- **API Integration**: RESTful API for programmatic access and automation

Chapter 12

Conclusion

The Product Review Sentiment Analysis system represents a comprehensive solution for automated customer feedback intelligence. By leveraging modern NLP techniques and interactive visualization frameworks, the system transforms the traditionally labor-intensive process of review analysis into an efficient, objective, and scalable operation. The four-layer architecture ensures maintainability and extensibility, while the intuitive dashboard interface makes sophisticated sentiment analysis accessible to non-technical stakeholders.

The system successfully addresses the core challenges of volume management, analysis speed, classification consistency, and insight quantification that plague manual review analysis approaches. Through its implementation of TextBlob sentiment analysis, Plotly visualizations, and Streamlit's interactive framework, the application delivers actionable intelligence that empowers businesses to make data-driven decisions regarding product quality, customer satisfaction, and brand perception.

As businesses continue to generate increasingly large volumes of customer feedback, automated sentiment analysis systems like this will become essential infrastructure for customer experience management and competitive intelligence operations. The foundation established by this project provides a robust platform for future enhancements and adaptations to evolving business intelligence requirements.

Chapter 13

References

Libraries and Frameworks

1. **Streamlit Documentation**: https://docs.streamlit.io/
2. **TextBlob Documentation**: https://textblob.readthedocs.io/
3. **PyPDF2 Documentation**: https://pypdf2.readthedocs.io/
4. **Pandas Documentation**: https://pandas.pydata.org/docs/
5. **Plotly Documentation**: https://plotly.com/python/

Academic References

1. Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.
2. Pang, B., & Lee, L. (2008). Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 2(1-2), 1–135.
3. Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4), 1093–1113.