FMDB *Transactions on Sustainable Finance and Data Science*

Explainable Agentic AI for Automated Customer Insight Generation from Product Reviews

Santosh Baral[1,*], Saroj Basyal[2], Shikha Jha[3], Shirsti Sharma[4], Mutaher Ahmed[5]

[1,2,3,4]Department of Computer Science and Technology, University of the West of Scotland, Glasgow, Scotland, United Kingdom.
[5]School of Computing and Artificial Intelligence, University of Law, London, England, United Kingdom.
baralsantos10@gmail.com[1], basyalsaroj001@gmail.com[2], shikhajha527@gmail.com[3], shrissharma049@gmail.com[4], mutaher.ahmed38@law.ac.uk[5]

Abstract: Customer reviews contain valuable information about customer experiences, product quality, and satisfaction. However, the increasing volume of online reviews makes manual analysis impossible. In this work, researchers design the Explainable Agentic AI for Automated Customer Insight Generation from Product Reviews system, which integrates NLP, sentiment analysis, topic modeling, XAI, and evidence-based customer insight generation into a unified analytics approach. The system was built in Python using the Jupyter Notebook framework and tested on an Amazon product review dataset of 72,500 reviews across five product categories. Researchers preprocessed and classified customer reviews using VADER, Logistic Regression, and Naïve Bayes. They used Latent Dirichlet Allocation (LDA) to identify discussion topics. To improve model transparency and explain sentiment predictions, researchers used SHAP and LIME. A deterministic, template-driven evidence synthesis process has generated structured customer insight reports from sentiment, topic, and explainability outputs. Researchers also implemented an interactive visualization dashboard. As a result of the experiment, Naive Bayes achieved the highest overall accuracy, while Logistic Regression achieved the highest Macro F1-score and was selected for the Explainable AI stage.

Keywords: Agentic AI; Sentiment Analysis; Customer Insight Generation; Five Product Categories; Latent Dirichlet Allocation; Explainable Agentic AI; Jupyter Notebook framework.

Received on: 21/12/2025, **Revised on:** 04/02/2025, **Accepted on:** 21/03/2026, **Published on:** 16/10/2026

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSFDS

DOI: https://doi.org/10.69888/FTSFDS.2026.000803

Cite as: S. Baral, S. Basyal, S. Jha, S. Sharma, and M. Ahmed, "Explainable Agentic AI for Automated Customer Insight Generation from Product Reviews," *FMDB Transactions on Sustainable Finance and Data Science,* vol. 1, no. 4, pp. 240–267, 2026.

Copyright © 2026 S. Baral *et al.,* licensed to Fernando Martins De Bulhão (FMDB) Publishing Company. This is an open access article distributed under CC BY-NC-SA 4.0, which permits copying, remixing, redistributing, transformation, and building upon the material in any medium so long as the original work is properly cited.

1. Introduction

With the rapid growth of e-commerce, customer reviews have become one of the most valuable sources of feedback for companies operating through e-commerce websites [1]. These reviews provide insights into customer experience, product quality, customer satisfaction, and buying behaviour, helping companies improve their products and services [2]. As a result, the explosion of online reviews has made manual analysis difficult [3]. AI and NLP offer an effective way to analyse vast

*Corresponding author.

amounts of unstructured text automatically [4]. This paper presents the research, introducing the background, research context, problem statement, research aim, objectives, research questions, rationale, scope, limitations, dissertation structure, and paper summary [5]. The rapid growth of e-commerce has generated vast amounts of customer reviews [6]. Customer reviews provide detailed information about customer experience, satisfaction, and product quality, helping companies make decisions [7]. Unlike ratings, textual reviews explain why customers feel the way they do [8]. Conventional approaches to manual review analysis have become impractical due to the sheer number of reviews produced each day [9]. As a result, AI and NLP approaches have become necessary to extract valuable information from large customer review datasets. Sentiment analysis determines whether a review is positive, negative or neutral, whereas topic modelling identifies the themes customers discuss [10]. Finally, recent innovations in XAI and Agentic AI increase transparency and automate analysis, enabling more trustworthy customer insights [11].

Innovations in AI have motivated organisations to move from purely predictive approaches to the development of explainable, trustworthy analytical systems [12]. Although numerous studies focus on one or several specific AI approaches, such as sentiment analysis and topic modelling, far fewer combine them into a single approach that can generate transparent customer insights [13]. On the other hand, businesses need AI-based systems that are not only efficient and provide valuable insights but can also be explained [14]. This research addresses these problems by proposing an Explainable Agentic AI approach to generating customer insights [15]. Despite the widespread use of sentiment analysis and topic modeling for customer reviews, these approaches have largely been used separately [16]. Sentiment analysis detects customer opinion without explaining why, while topic modeling identifies discussion topics without indicating whether customers feel positively or negatively about them [17]. Moreover, most AI models act like black boxes because they don't explain how they make predictions [19]. Current solutions also require significant manual work to interpret their analytics [18]. Thus, an integration framework is needed to provide users with explainable, automated customer insights from product reviews [20].

1.1. Research Aim

This research aims to create and assess an Explainable Agentic AI framework for automatically generating customer insights from product reviews [21].

1.2. Research Objectives

The objectives of the research include:

- Preprocessing customer review data using NLP techniques.
- Performing sentiment analysis using machine learning algorithms.
- Discovering underlying topics using Latent Dirichlet Allocation (LDA).
- Increasing model transparency using SHAP and LIME.
- Creating an Agentic AI layer to generate customer insights automatically.
- Evaluating the proposed framework both quantitatively and qualitatively.

1.3. Research Questions

The following research questions are addressed:

- How can NLP be used to preprocess product reviews for customer insight generation?
- How effective is the integration of sentiment analysis and topic modelling for customer insight generation?
- How can Explainable AI be used to make customer insight generation transparent and trustworthy?
- How can the Agentic AI framework be used to generate customer insights automatically?

With the increasing volume of online customer reviews, there is a need to develop automated systems that can obtain valuable and reliable information about the business. Although some studies show that sentiment analysis and topic modeling can be applied effectively separately, the literature lacks a single framework that integrates these methods with Explainable AI and Agentic AI [23]. Therefore, this research aims to fill this gap by developing a framework that uses multiple AI methods to generate valuable customer insights [22]. The research scope includes English-language Amazon product reviews collected from a publicly available Kaggle dataset. The framework includes data preprocessing, sentiment analysis, topic modelling, Explainable AI, and automated customer insights. The research will apply classical machine learning algorithms and present the results via an interactive dashboard. This research does not consider multilingual datasets, social media reviews, real-time data processing, or agentic AI based on LLMs for autonomous reasoning. The research is limited to a single Amazon product review dataset, which may reduce the generalizability of the findings to other platforms and domains. Manual interpretation of

topic labels is required because the LDA model cannot automatically provide meaningful topic names. The Agentic AI layer also uses deterministic, template-based reasoning and does not autonomously use LLM-based reasoning.

2. Literature Review

The rapid growth of e-commerce has produced a large volume of online customer reviews that serve as a source of business intelligence. Reviews provide extensive information about customers' experiences, opinions, and expectations; however, their sheer volume makes manual analysis difficult. As a result, more and more organizations are using artificial intelligence and natural language processing to analyze customer feedback. Recent developments in NLP have made automatic analysis, such as sentiment analysis and topic modelling, possible, allowing identification of customers' opinions and the topics discussed in reviews. Nevertheless, most of these approaches focus only on prediction, without explaining how they produce these insights [1]. This reduces user trust and makes practical AI use in business environments difficult. Explainable artificial intelligence (XAI) can address low interpretability, while agentic AI enables the integration of analytical tools and automates the generation of customer insights. The paper also describes shortcomings in current research. It presents the research gap the dissertation seeks to fill: creating a framework for Explainable Agentic AI to generate customer insights from Amazon product reviews.

2.1. Online Customer Reviews and Customer Insight Generation

The rapid growth of e-commerce has made online customer reviews an important source of business intelligence, providing insights into customer experiences, product quality, and satisfaction. Unlike numerical ratings, textual reviews explain the reasons behind customer opinions, making them more valuable for understanding customer behaviour. However, the growing volume of reviews makes manual analysis impractical, and informal language and inconsistent writing styles further challenge automated analysis [2]. Consequently, Artificial Intelligence (AI) and Natural Language Processing (NLP) techniques are widely used to extract meaningful insights. Previous studies show that combining sentiment analysis with topic modelling provides a more comprehensive understanding of customer opinions than either approach alone.

2.2. Natural Language Processing in Customer Review Analysis

Natural Language Processing (NLP) enables the analysis of unstructured text and has become essential for extracting valuable information from online customer reviews. However, customer reviews may include many non-standard linguistic phenomena, including typos, slang, abbreviations and other informal language. Hence, text pre-processing became an inevitable step before text analysis [3]. The literature mentions preprocessing methods such as tokenization, stop word removal, lemmatization and text normalization. TF-IDF can represent preprocessed text numerically, highlighting significant terms while reducing the impact of common words.

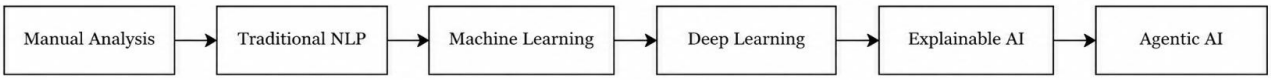


Figure 1: Evolution of customer review analysis techniques

Figure 1 shows how customer review analysis has developed over time. Early approaches involved manual review, followed by traditional NLP and machine learning techniques. More recent innovations include Explainable AI and Agentic AI. Traditional machine learning algorithms, such as Logistic Regression and Naive Bayes models, remain widely used because they provide efficient, interpretable text classification at lower cost than deep learning algorithms [4]. However, while being less accurate than deep learning models, traditional models are more interpretable and computationally economical. Therefore, this dissertation is based on traditional NLP techniques with Explainable AI:

$$TF - IDF(t,d) = TF(t,d) \times log(\frac{N}{DF(t)}) \quad (1)$$

TF(t,d) is the frequency of term t in document d, DF(t) is the number of documents where the term occurs, and N is the total number of documents. The TF-IDF algorithm assigns more weight to discriminative terms while giving less importance to frequently appearing terms.

2.3. Sentiment Analysis of Customer Reviews

Opinion mining or sentiment analysis is an NLP approach to classifying customers' sentiments as positive, negative or neutral to understand client satisfaction and inform business decisions. Sentiment analysis uses lexicon- and machine learning-based

approaches. The lexicon approach relies on predefined sentiment dictionaries and is computationally efficient but cannot address context, sarcasm or domain-specific language. Machine learning models, in turn, learn sentiment patterns from labeled datasets and, therefore, provide better classification performance [5]. Logistic regression and Naïve Bayes are classical classifiers that remain in use because they process high-dimensional textual data efficiently while consuming relatively few resources. In contrast, transformer models like BERT achieve high precision because they capture word relationships and context in text. However, transformer-based models are less interpretable and require more computational resources than machine learning-based models:

$$P(y = 1 \mid x) = \frac{1}{1+e-(wTx+b)} \quad (2)$$

Logistic regression estimates the probability that a review belongs to a particular sentiment class using the sigmoid function, producing probabilities between 0 and 1. While transformer-based architectures like BERT achieve better classification performance, traditional machine learning methods are still preferred because they require less computation, train faster and are more compatible with explainability methods such as SHAP.

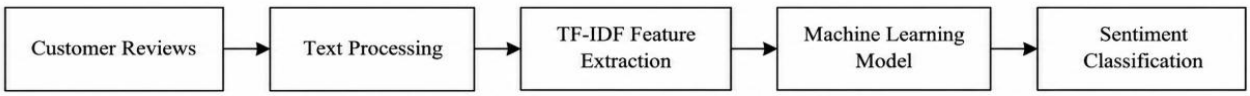


Figure 2: NLP-based customer review analysis pipeline

Figure 2 shows an NLP pipeline for analysing customer reviews. It illustrates how the review text is preprocessed and features are extracted before machine learning algorithms analyse it. Despite its effectiveness, sentiment analysis alone cannot explain why customers hold opinions [6]. Therefore, many studies recommend combining sentiment analysis with topic modelling to identify both customer sentiment and the underlying themes discussed in reviews. In this dissertation, sentiment analysis uses traditional machine learning models, and the results are integrated with topic modelling and Explainable AI to generate transparent customer insights.

2.4. Topic Modelling for Customer Review Analysis

Arrieta et al. [3] argue that topic modelling is an unsupervised NLP methodology for discovering latent topics in large amounts of unlabelled data. Topic models differ from supervised learning methodologies in that they cluster commonly occurring words into relevant topics; therefore, they suit large-scale analyses of customer reviews, where manual classification would be challenging. The Latent Dirichlet Allocation (LDA) is one of the most widely applied topic models. Each document is represented as a combination of multiple topics, with each topic defined as a probability distribution over words. In customer reviews, LDA is widely used to identify recurring themes such as product quality, price, delivery, usability, and customer support, helping the company understand customer issues and improve its products or services. However, topic modelling success depends heavily on text preprocessing, including tokenisation, stop word removal, lemmatisation, and other noise-reduction techniques. Moreover, choosing the optimal number of topics matters, since too few topics may merge several themes into one. At the same time, too many topics can split valuable information across several themes. Although BERTopic, along with other transformer-based methods, provides a more comprehensive semantic representation, LDA remains widely used because it computes efficiently and offers an interpretable probability model. Nevertheless, LDA has drawbacks despite its widespread use. The topics it generates are hard to interpret because the algorithm produces groups of statistically related words without labels. In addition, LDA is difficult to apply to short texts, sophisticated languages, and word-level semantic relations because it relies on word co-occurrence. Transformer models are better at semantic processing than the discussed method; however, they usually require more computing power and produce harder-to-interpret topic structures. Thus, LDA remains a convenient method for exploratory customer reviews analysis because of its low computational complexity and interpretability:

$$P(w \mid d) = z = \sum_{z=1}^{K} P\,(w \mid z)P(z \mid d) \quad (3)$$

Where P(w|z) is the probability of a word given a topic, and P(z|d) is the probability of a topic within a document. In this dissertation, LDA will identify the major topics in Amazon product reviews. The analysis will use the most representative keywords from the discovered topics, along with sentiment analysis.

2.5. Explainable Artificial Intelligence and Agentic AI

The growing use of Artificial Intelligence (AI) in business decision-making has highlighted the need for transparency and interpretability. Although machine learning models often achieve high predictive performance, many operate as "black boxes",

making it difficult for users to understand how predictions are generated. Explainable Artificial Intelligence (XAI) addresses this challenge by providing understandable explanations of model decisions. Lundberg and Lee [23] note that common XAI techniques include Local Interpretable Model-Agnostic Explanations (LIME), which explains individual predictions, and Shapley Additive explanations (SHAP), which measures each feature's contribution to a prediction. These methods improve user trust and make AI models more suitable for business applications. Alongside XAI, Agentic AI has emerged as an approach for coordinating multiple analytical tasks within a single workflow. In this dissertation, Agentic AI is implemented as an evidence-orchestration framework that integrates the outputs of sentiment analysis, topic modelling, and Explainable AI using template-based reasoning to generate structured customer insight reports. This approach improves transparency and reproducibility while laying the groundwork for future LLM integration:

$$f(x) = \phi_0 + i = \sum_{i=1}^{M} \phi_i \quad (4)$$

where ϕi represents the contribution of each feature to the final prediction, allowing transparent interpretation of model decisions. SHAP provides theory-driven consistency for both local and global explanations, but it is computationally more costly than LIME. LIME is faster, but it may be less stable because it relies on local approximations.

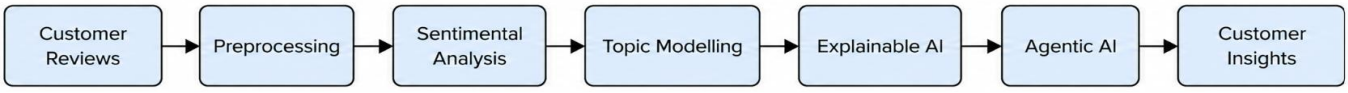


Figure 3: Integrated explainable agentic AI framework

Combining Explainable AI with Agentic AI ensures that automatically generated customer insights remain transparent, interpretable, and suitable for business decision-making. Figure 3 shows how sentiment analysis, topic modelling, Explainable AI, and Agentic AI work together in a single analytical process. Using these approaches, researchers can create structured, evidence-based insights from customers' reviews.

2.6. Research Gap

Several findings indicate that NLP tools such as sentiment analysis and topic modelling can be used effectively in customer review analysis. Nevertheless, there are several research gaps related to customer review analysis: First, almost all existing studies treat sentiment analysis and topic modelling separately, without integrating them into a single framework. Sentiment analysis reveals customers' opinions; however, it does not explain why. Topic modelling, by contrast, helps identify discussion themes but cannot determine customer sentiment toward them. Second, most existing studies focus on improving prediction accuracy while ignoring model interpretability, which reduces users' trust in models and limits the use of AI in business decision-making [7]. Thus, there is a necessity to use Explainable AI in customer review analysis. Finally, current customer review analysis systems need manual interpretation of analytical outcomes. Despite the automation opportunities offered by Agentic AI, research in this area remains limited. The current literature has mostly assessed analysis methods individually. Little work has examined combining sentiment analysis, topic modeling, explainability, and evidence orchestration into a single framework. This paper bridges this gap by introducing an Explainable Agentic AI framework that integrates these complementary techniques into a single pipeline for generating customer insights.

3. Research Design

This paper describes the methodology used to design, develop, and evaluate the proposed Explainable Agentic AI framework for automatically generating customer insights from Amazon product reviews. Unlike the project specification, which defines the project goals, this paper details the methods and procedures used to implement and evaluate the proposed solution. The research adopts the Design Science Research (DSR) methodology because its main goal is to design and evaluate an AI-based artefact to address the problem of generating explainable customer insights. This paper covers the research philosophy, research approach, dataset selection, data preparation, natural language processing techniques, sentiment analysis, topic modeling, explainable AI, agentic AI implementation, and evaluation methodologies. The general methodology includes data collection and preprocessing, applying NLP techniques, sentiment analysis, topic modeling, explainability analysis, automated customer insight generation, and evaluation.

3.1. Research Philosophy

This research uses a pragmatic philosophy, as developing an Explainable Agentic AI requires both quantitative and qualitative approaches. The quantitative aspect of the research involves measuring machine learning model performance using indicators such as accuracy, precision, recall, and F1-score. It is necessary to evaluate the performance of sentiment analysis and topic

modelling techniques. Moreover, generating and evaluating customer insights involves subjective judgments about the usefulness, clarity and interpretability of the explanations provided to users. Thus, the positivist philosophical approach does not cover all research purposes, as it focuses only on measurable results. In addition, the interpretivist philosophical approach is not suitable because it requires an objective evaluation of AI model performance. Pragmatic philosophy allows researchers to use methods best suited to the research purposes, rather than strictly following a single philosophical paradigm. Pragmatist philosophy suits the present research because it combines machine learning techniques with explainability and AI insight generation. The limitation of pragmatist philosophy, which is its possible orientation to the practical side rather than philosophy, is known. The chosen approach provides a balanced framework for developing and evaluating an AI artefact that aims to produce accurate, transparent, and useful customer insights.

3.2. Research Approach

This dissertation employs an abductive research approach that moves back and forth between existing theories and practical observations throughout the process of developing an AI system. Unlike deductive reasoning, which starts by verifying a hypothesis, abductive reasoning seeks the best possible explanation or solution by combining existing theories with practical observations. Induction, by contrast, builds theories from collected data. During this research, an iterative approach is adopted in which appropriate methods are analyzed, applied, tested, and improved based on their effectiveness and results. In other words, this research examines text preprocessing approaches, sentiment analysis techniques, and topic modeling approaches to identify methods for deriving valuable information from customer Amazon reviews. Thus, an abductive approach supports this research's exploratory, artefact-developing nature by enabling interaction between theory and practice throughout the development process.

3.3. Research Strategy: Design Science Research

The Design Science Research (DSR) approach was chosen because this research aims to create and evaluate an Explainable Agentic AI framework that automatically generates customer insights from Amazon product reviews. Unlike surveys and case studies, DSR emphasizes designing, implementing, and evaluating an artefact that solves a real-world problem. The artefact includes components such as NLP, sentiment analysis, topic modelling, XAI, and Agentic AI, enabling explainable customer insights. The DSR method offers an approach to artefact development and evaluation, but artefact validation was conducted using only one public Amazon review dataset.

3.4. Dataset Description

For this research, an Amazon product review dataset serves as the data source for developing and evaluating the Explainable Agentic AI framework. This dataset includes customer reviews that capture customer experiences, product quality and satisfaction levels. Customer reviews were chosen because they offer rich content that can be analyzed using NLP techniques to extract customers' opinions and trends. First, the dataset consists of about 72,500 product reviews with attributes such as review ID, review date, product category, product name, review title, review rating, review text, and review consistency. Of these attributes, the review text is the source of information for NLP analysis, and product-related attributes help provide additional customer insights. This dataset suits this research because it includes both numerical and textual customer feedback. Review ratings support sentiment classification, while review text offers additional insights through sentiment analysis and topic modeling. Additionally, product information can provide insights into specific product types, rather than only analyzing customers' opinions. The dataset is public and contains no personally identifiable information, so privacy is not a concern.

3.5. Data Collection and Preparation

This research used secondary data collected from an open-source Amazon product review dataset on Kaggle. This data type was chosen because secondary data was necessary to create an AI-based analysis framework. The dataset contains many customer reviews, making it well-suited for NLP and machine learning analysis. Before building the model, researchers performed several procedures to improve data quality and prepare the review text for further analysis. This preparation process included data cleaning, text normalization, tokenization, stop word removal and lemmatization. These methods reduce noise, normalize the text data and convert customer reviews into the appropriate form. This prepared dataset was used for sentiment analysis, topic modeling, Explainable AI and Agentic AI to provide structured customer insights.

3.6. Research Workflow and System Design

This flowchart represents the entire research procedure in this paper, from dataset preparation to generating explainable customer insights. This proposed framework incorporates natural language processing (NLP), machine learning, XAI, and agentic AI methodologies to analyze Amazon product reviews (Figure 4).

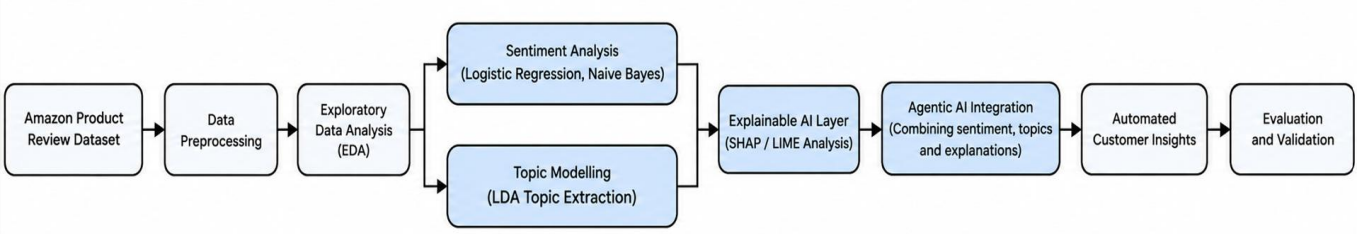


Figure 4: Workflow diagram of the project

3.7. Workflow Description

- **Dataset Input:** The study uses a publicly available Amazon product review dataset containing customer reviews and product information. The dataset provides the text needed for sentiment analysis, topic modelling, and generating customer insights.
- **Data Preprocessing:** The review data is preprocessed through data cleaning, text normalisation, tokenisation, stop-word removal, and lemmatisation to improve data quality and prepare the text for analysis.
- **Exploratory Data Analysis:** Exploratory Data Analysis (EDA) examines the distribution of reviews, ratings, and other dataset characteristics before model development.
- **Sentiment Analysis:** Machine learning models, including Logistic Regression and Naïve Bayes, classify reviews as positive, negative, or neutral. Model performance is evaluated using accuracy, precision, recall, and F1-score.
- **Topic Modelling:** Latent Dirichlet Allocation (LDA) identifies the main discussion topics and extracts representative keywords from customer reviews.
- **Explainable AI Layer:** SHAP and LIME explain model predictions by identifying the features that most influence sentiment classification.
- **Agentic AI Layer:** The Agentic AI framework orchestrates the outputs of sentiment analysis, topic modelling, and Explainable AI. An evidence-gathering stage collects structured analytical results, which a template-based reasoning process synthesises into customer insight reports.
- **Insight Generation and Evaluation:** The generated customer insights are evaluated by assessing model performance, topic relevance, explainability, and report consistency.

3.8. NLP and Machine Learning Techniques

This study uses NLP and Machine Learning to analyze customer reviews and extract useful insights. The approach includes Sentiment Analysis, Topic Modelling, Explainable AI, and Agentic AI:

- **Sentiment Analysis:** Sentiment analysis categorizes customer reviews as positive, negative, or neutral using TF-IDF and conventional machine learning techniques such as logistic regression and Naive Bayes. The model performance is measured using accuracy, precision, recall, F1-score, and the confusion matrix.
- **Topic Modelling:** Topic modelling identifies hidden themes in customer reviews without predefined categories. This study applies Latent Dirichlet Allocation (LDA) to extract dominant topics and representative keywords from preprocessed reviews. The study combines the extracted topics with sentiment analysis to generate comprehensive customer insights.
- **Explainable Artificial Intelligence (XAI):** Explainable AI (XAI) improves the transparency of machine learning predictions. This study applies SHAP and LIME to identify the features influencing sentiment classification, enabling users to understand model decisions and increasing trust in the generated customer insights.
- **Agentic AI Integration:** In the Agentic AI layer, the outputs of sentiment analysis, topic modeling, and Explainable AI are integrated into the evidence-gathering and template reasoning process. This approach automatically generates customer insight reports based on Amazon product reviews.
- **Evaluation Framework:** The Explainable Agentic AI framework is assessed using both quantitative and qualitative methods. Considering that there are several tools employed within Explainable Agentic AI, such as machine learning models, topic modelling algorithms, explainability approaches, and AI-generated insights, several ways of assessing different parts of the framework need to be applied.
- **Sentiment Analysis Assessment:** Sentiment analysis algorithms are tested using an 80-20 train-test split. The model's performance is evaluated using accuracy, precision, recall, F1 score, and a confusion matrix to compare classification across sentiment categories.

- **Topic Modelling Evaluation:** Topic modelling is evaluated by assessing the coherence and interpretability of the extracted topics. Representative keywords are examined to determine whether the identified topics reflect meaningful customer concerns and discussion themes. Beyond coherence scores, the analysis examines important keywords in each topic to determine whether they represent comprehensible customer concerns, such as product quality, usability, price, or performance.
- **Explainability Evaluation:** The explainability module is evaluated according to whether generated explanations contain valuable information related to model decisions. Explanations are checked to see whether important features identified by XAI methods match the content of the original customer reviews.

The final stage of insight generation is evaluated based on whether generated insights accurately reflect the outputs of sentiment analysis and topic modelling processes. The customer insights reports created through the process are checked to ensure they accurately depict the output from sentiment analysis, topic modelling, and Explainable AI. Because customer insights reports are created using templates, the team manually checks faithfulness by matching the generated reports with the original customer reviews.

3.9. System Overview Assessment

The entire system is assessed based on its capabilities to:

- Find valuable patterns in customers' reviews.
- Integrate various AI methods into a single process.
- Provide understandable reasoning behind generated information.
- Generate valuable results for analyzing customers' attitudes.

This methodology allows assessment of the system not only for accuracy but also for interpretability, robustness, and practical applicability.

3.10. Validity, Reliability, and Limitations of the Research Design

The framework was evaluated using both quantitative model performance and qualitative customer insights to support credible and reproducible results.

3.10.1. Validity Considerations

- **External Validity:** The study used a public Amazon review dataset, so findings may not generalise to other platforms, products, or languages.
- **Construct Validity:** Sentiment, topic relevance, and explanation quality were assessed using consistent evaluation criteria, although some interpretation involved human judgement.
- **Technical Validity:** Appropriate performance metrics and validation methods supported reliable results.
- **Reliability Issues:** The study's reliability was supported by reliable NLP and machine learning techniques, along with consistent preprocessing, modelling, and evaluation methods, enabling replication. The study assessed insights using consistent standards. Future work may increase reliability by using multiple independent assessors.
- **Research Limitations:** The study's key limitations were its reliance on a single Amazon review dataset, the subjectivity of human interpretation of LDA topics, and the use of deterministic, template-based reasoning rather than LLM-based reasoning. Although the deterministic approach increases reliability, its limited flexibility can be addressed in future work.

4. Implementation

This paper provides a detailed description of the explainable agentic AI system implemented to automatically generate customer insights from Amazon product reviews. This paper details how each architectural component was implemented, explains the rationale behind key design choices, and describes challenges encountered during development.

This paper describes the pipeline implementation, which consists of the following steps: data preprocessing, exploratory data analysis, sentiment classification, explainability, topic modeling, agentic evidence synthesis, and dashboard creation. Each step is handled by the relevant team member, but the pipeline design ensures each component depends on the previous step's output through a common dataset.

4.1. Tools Used

The entire paper was written in Python using Jupyter Notebook. Main tools (Table 1):

Table 1: Tools used for project

Tool / Library	Purpose
Python	System implementation
Jupyter Notebook	Development environment
pandas, numpy	Data manipulation and analysis
nltk, re	Text preprocessing
scikit-learn	Machine learning and topic modelling
vaderSentiment	Lexicon-based sentiment analysis
SHAP, LIME	Explainable AI
Plotly, Matplotlib, Seaborn	Data visualisation
ipywidgets	Interactive dashboard

All intermediate outputs were saved in a single panda DataFrame (df), and new columns were added at each step to facilitate efficient reuse and minimize duplication (Figure 5).

```
import numpy as np
import pandas as pd
import re
import string
import matplotlib.pyplot as plt
import seaborn as sns

# NLP
nltk.download('stopwords', quiet=True)
nltk.download('punkt', quiet=True)
nltk.download('punkt_tab', quiet=True)
nltk.download('wordnet', quiet=True)
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer

# Sentiment
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

# ML
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split, StratifiedKFold, cross_val_score
from sklearn.metrics import (classification_report, accuracy_score,
                             confusion_matrix, f1_score)

# SHAP
import shap

print("All libraries loaded successfully.")
print(f"SHAP version: {shap.__version__}")
```

Figure 5: Libraries used in project

4.2. Data Collection and Cleaning

The Amazon Product Reviews dataset was selected because it contains real customer feedback from multiple product categories, making it suitable for sentiment analysis, topic modelling, and automated insight generation. The dataset has 72,500 Amazon reviews across 5 departments. Each review includes title, text, star rating, product name, and date.

4.2.1. Cleaning Steps

- Removed missing/empty reviews
- Removed duplicate reviews
- Flagged reviews where the rating and text didn't match (kept instead of deleted) (Figure 6).

```
print("=== Missing Values ===")
print(df.isnull().sum())
print("\n=== Star Rating Distribution ===")
print(df['reviewStar'].value_counts().sort_index())
print("\n=== Departments ===")
print(df['mainDepartment'].value_counts())
print("\n=== Inconsistent Reviews ===")
print(df['inconsistentStatus'].value_counts())
```

```
=== Missing Values ===
reviewId              0
reviewDate            0
mainDepartment        0
subDepartment         0
productName           0
reviewTitle          20
reviewStar            0
reviewText            7
inconsistentStatus    0
dtype: int64
```

Figure 6: Snapshot of implementing missing values

4.3. Exploring the Data

Basic checks: how ratings are distributed, how many reviews each department has, and review length (Figure 7).

```
df = df.dropna(subset=['reviewText', 'reviewTitle'])
df = df.drop_duplicates(subset=['reviewId'])
df = df.reset_index(drop=True)
print(f"After:  {df.shape[0]} rows")

# Combine title + review text
df['combined_text'] = df['reviewTitle'].astype(str) + ' ' + df['reviewText'].astype(str)

# Review length
df['review_length'] = df['reviewText'].apply(lambda x: len(str(x).split()))
print("\nReview length statistics:")
print(df['review_length'].describe())

plt.figure(figsize=(10, 4))
plt.hist(df['review_length'], clip(upper=300), bins=50, color='steelblue', edgecolor='white')
plt.title('Review Length Distribution (clipped at 300 words)')
plt.xlabel('Word Count')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```

Figure 7: Snapshot of implementing bar chart

The Exploratory Data Analysis (EDA) was performed to understand the dataset's properties before model construction. It involved examining the distribution of review ratings, review frequency by department and review length (Figure 8).

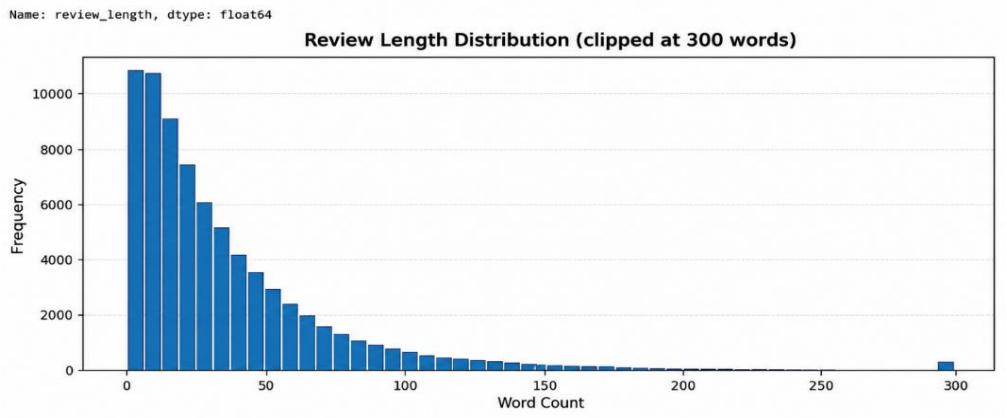


Figure 8: Bar chart according to word count and frequency

Researchers also extracted the review date from the original review text using regular expressions to analyze customer review trends over time (Figure 9).

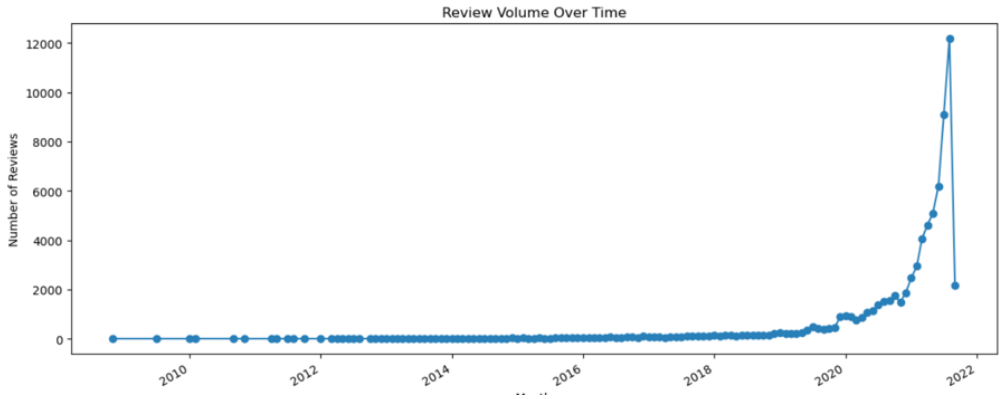


Figure 9: Snapshot of review volume over time

Date parsing: dates were stored inside a sentence (e.g., "Reviewed in the United States on August 8, 2021") (Figure 10).

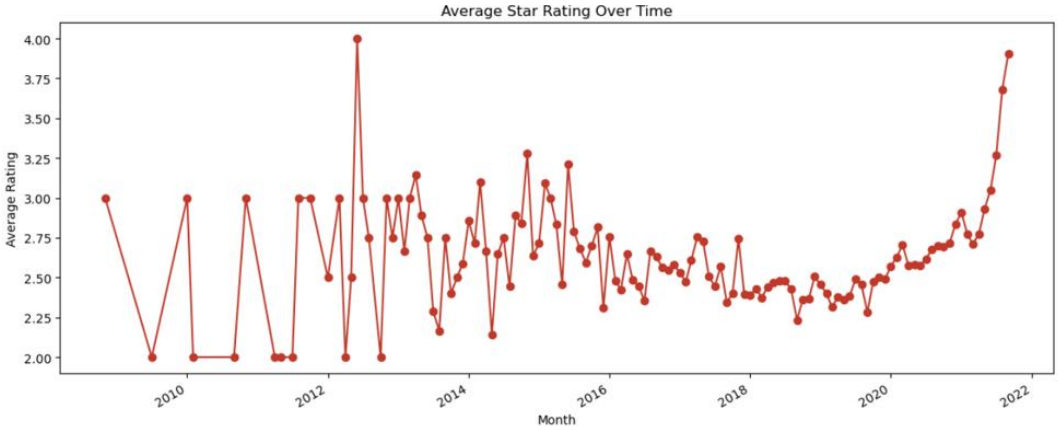


Figure 10: Snapshot of average star rating over time

A regex pulled out the date so researchers could track trends over time (Figure 11).

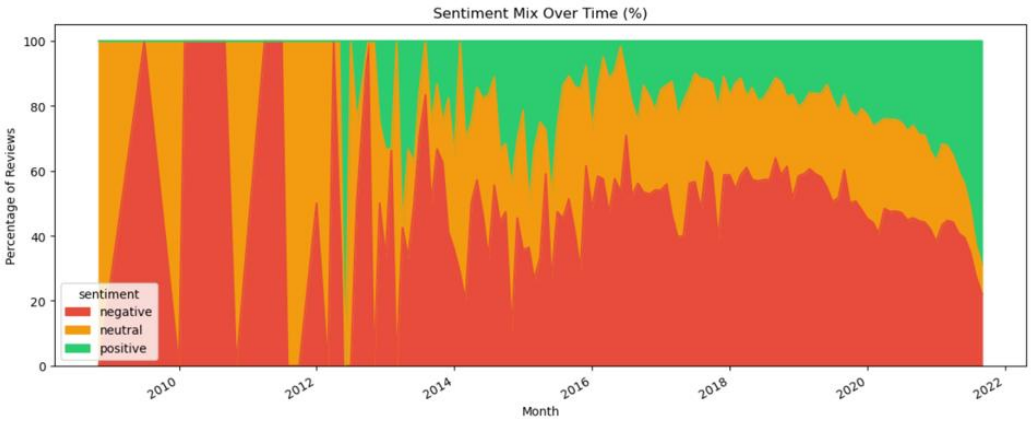


Figure 11: Snapshot of sentiment analysis in percentage

4.4. Sentiment Analysis

Three approaches were compared, all evaluated on the same 3-class problem (Figure 12):

```python
def star_to_sentiment(star):
    if star <= 2:   return 'negative'
    elif star == 3: return 'neutral'
    else:           return 'positive'

df['sentiment'] = df['reviewStar'].apply(star_to_sentiment)

print("Sentiment Distribution:")
print(df['sentiment'].value_counts())
print(f"\nClass balance (%):")
print((df['sentiment'].value_counts(normalize=True) * 100).round(2))
```

Figure 12: Snapshot of implementing sentiment analysis

For both supervised models, we trained on TF-IDF features from the preprocessed review data, using a vocabulary of 20,000 and unigrams and bigrams (Table 2).

Table 2: Comparing models with accuracy and F1

Model	Accuracy	Macro F1
VADER (raw text)	0.591	0.488
Logistic regression (preprocessed)	0.706	0.666
Naive Bayes (preprocessed)	0.721	0.584

The train-test split is 80:20, resulting in 57,980 training samples and 14,495 test samples (Figure 13).

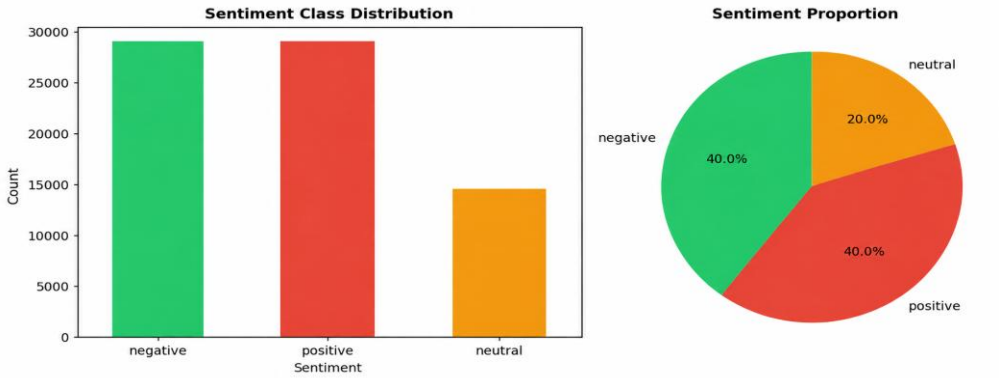


Figure 13: Snapshot of sentiment analysis

Although Naive Bayes achieved the best overall accuracy (72.1%), Logistic Regression had the best macro F1-score (0.666) (Figure 14).

```python
print("VADER BASELINE EVALUATION (on raw text)")
print("=" * 55)
print(f"Accuracy: {accuracy_score(df['sentiment'], df['vader_sentiment']):.4f}")
print(f"Macro F1: {f1_score(df['sentiment'], df['vader_sentiment'], average='macro'):.4f}")
print()
print(classification_report(df['sentiment'], df['vader_sentiment'],
                            target_names=['negative', 'neutral', 'positive']))

cm_vader = confusion_matrix(df['sentiment'], df['vader_sentiment'],
                            labels=['negative', 'neutral', 'positive'])
plt.figure(figsize=(7, 5))
sns.heatmap(cm_vader, annot=True, fmt='d', cmap='Blues',
            xticklabels=['negative', 'neutral', 'positive'],
            yticklabels=['negative', 'neutral', 'positive'])
plt.title('VADER Confusion Matrix (raw text input)')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.tight_layout()
plt.show()
```

Figure 14: Snapshot of implementing VADER confusion matrix

Therefore, Logistic Regression was chosen as the classifier for the Explainable AI part. Further, five-fold cross-validation confirmed the chosen model, yielding an average macro F1-score of 0.671 (±0.006) (Figure 15).

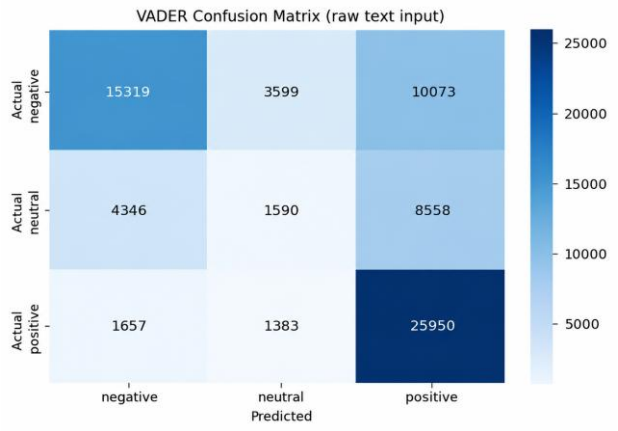


Figure 15: Snapshot of VADER confusion matrix

Sentiment labels were derived from star ratings using a three-class mapping: 1–2 stars (negative), 3 stars (neutral), and 4–5 stars (positive) (Figure 16).

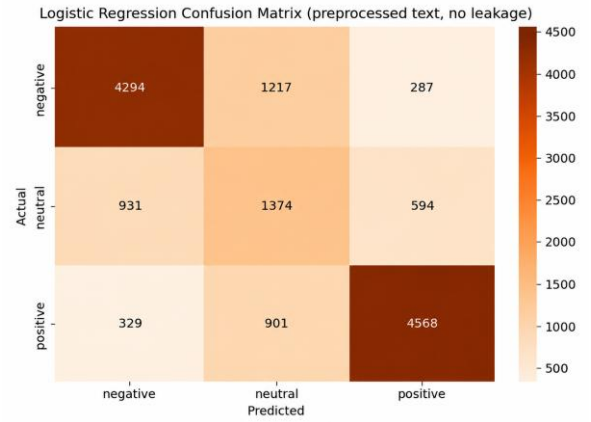


Figure 16: Logistic regression confusion matrix

This approach better reflects the ambiguity of 3-star reviews than binary classification and provides more meaningful business insights. The resulting label distribution was 40% negative, 20% neutral, and 40% positive (Figure 17).

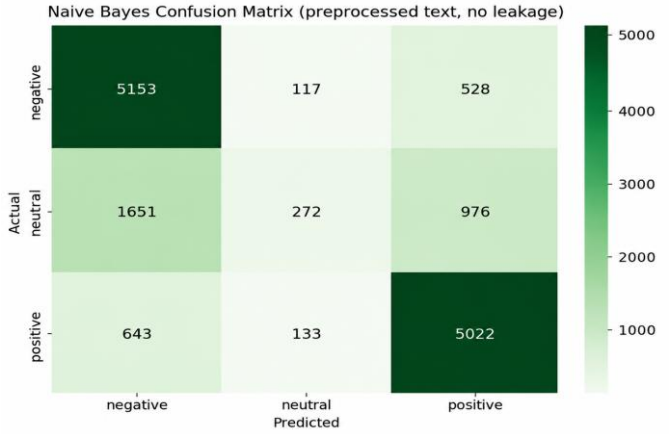


Figure 17: Naive Bayes confusion matrix

4.5. Explaining the Model's Decisions

Instead of using a single explainability approach, researchers used two separate approaches: SHAP and LIME (Figure 18).

```
fig, axes = plt.subplots(1, 3, figsize=(18, 6))

for idx, cls in enumerate(lr_model.classes_):
    coef        = lr_model.coef_[idx]
    top_pos_idx = coef.argsort()[-n_top::-1]
    top_neg_idx = coef.argsort()[:n_top]
    all_words   = ([feature_names[i] for i in top_pos_idx] +
                   [feature_names[i] for i in top_neg_idx])
    all_scores  = [coef[i] for i in top_pos_idx] + [coef[i] for i in top_neg_idx]
    colors      = ['#2ecf71' if s > 0 else '#e74c3c' for s in all_scores]

    axes[idx].barh(range(len(all_words)), all_scores, color=colors, alpha=0.8)
    axes[idx].set_yticks(range(len(all_words)))
    axes[idx].set_yticklabels(all_words, fontsize=9)
    axes[idx].axvline(x=0, color='black', linewidth=0.8)
    axes[idx].set_title(f'Top Features — {cls.upper()}', fontweight='bold')
    axes[idx].set_xlabel('Coefficient Weight')

plt.suptitle('Global Explainability: TF-IDF Feature Weights per Sentiment Class',
             fontsize=13, fontweight='bold')
plt.tight_layout()
plt.show()
```

Figure 18: Snapshot of implementing TF-IDF

Using two approaches was intentional to strengthen the validity of the explainability layer, as SHAP and LIME operate on different principles (SHAP uses game theory and Shapley values, while LIME approximates local model behavior with a surrogate model built on perturbed inputs) (Figure 19).

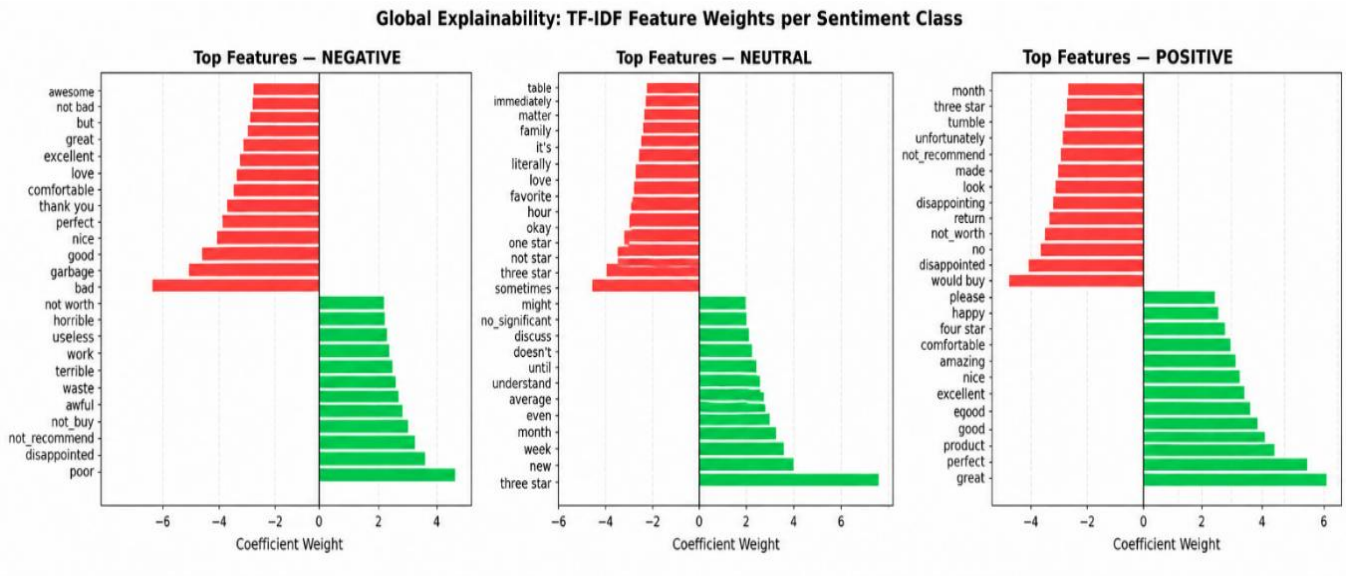


Figure 19: TF - IDF feature weights

Therefore, consistency between the two methods provides much stronger evidence of feature importance than using only one approach (Figure 20).

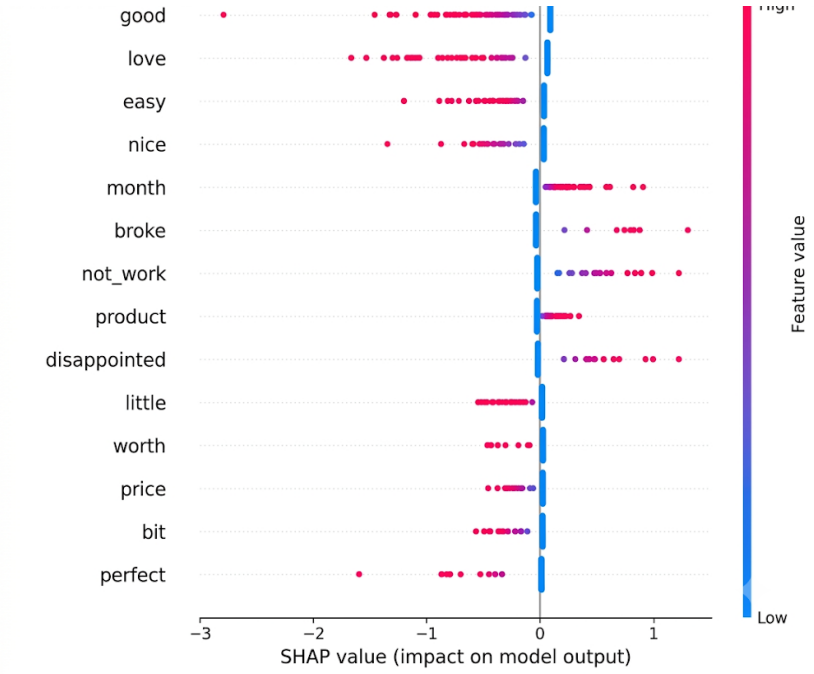


Figure 20: Snapshot of value using SHAP – negative

Used two tools to explain why the model predicted a sentiment: SHAP – shows which words mattered most, overall and per review (Figure 21).

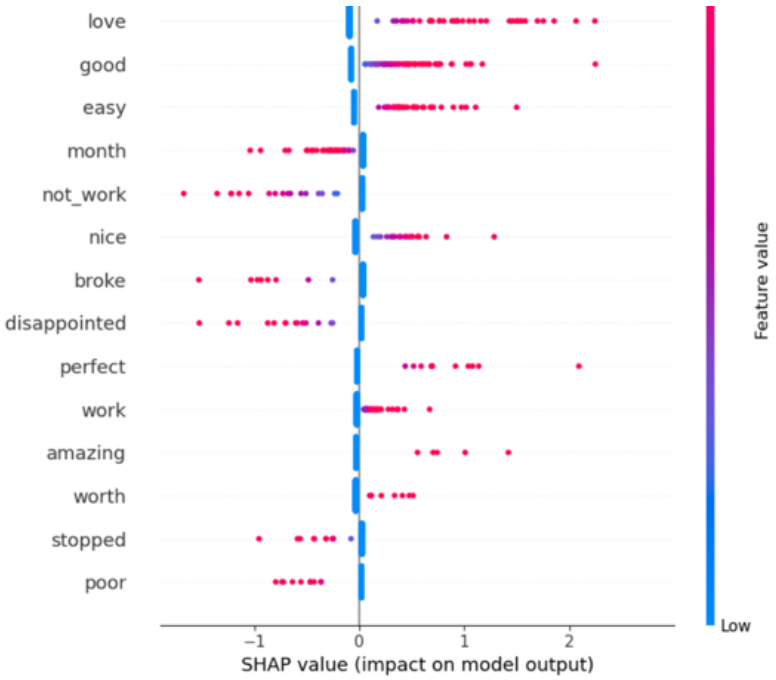


Figure 21: Snapshot of SHAP values – positive

The Python code is for sentiment analysis and model interpretation using SHAP (SHapley Additive exPlanations). The system uses a trained Logistic Regression model to classify customer reviews as positive, negative or neutral. The code compares the true sentiment with the predicted sentiment and selects correctly classified reviews for each category. It then shows the predicted sentiment with the original review text. Finally, it produces a SHAP force plot that explains how individual words contribute to the model's prediction. This identifies the important words affecting sentiment categorization and helps explain the machine learning model's decision-making (Figure 22).

```
preds = lr_model.predict(X_shap_sparse)

for target_class in ['positive', 'negative', 'neutral']:
    class_idx = list(lr_model.classes_).index(target_class)
    mask      = (y_shap_labels == target_class) & (preds == target_class)

    if mask.sum() == 0:
        print(f"No correctly predicted {target_class} example found.")
        continue

    example_idx       = np.where(mask)[0][0]
    original_text_idx = sample_idx[example_idx]
    original_text     = df['combined_text'].iloc[original_text_idx]

    print(f"--- {target_class.upper()} review ---")
    print(f"Original:  {original_text[:200]}")
    print(f"Predicted: {target_class}\n")

    shap.force_plot(
        explainer.expected_value[class_idx],
        shap_values[class_idx][example_idx],
        feature_names=feature_names,
        matplotlib=True,
        show=True,
        figsize=(20, 3)
    )
    print()
```

Figure 22: Snapshot of implementing SHAP using sentiment analysis

The above image is a SHAP force plot. It shows how different words influence a machine learning model's sentiment prediction for a customer review. The final output value is 1.14, and the base value is the model's first prediction. The pink color shows words that push the prediction upwards, such as “right mattress”, “soaked,” and “not waterproof”. The blue color indicates words that reduce the prediction, such as “wonderful.” This helps us identify which words contribute positively or negatively to the model output. This visualization helps us understand how the model interprets customer reviews and makes its predictions (Figure 23).

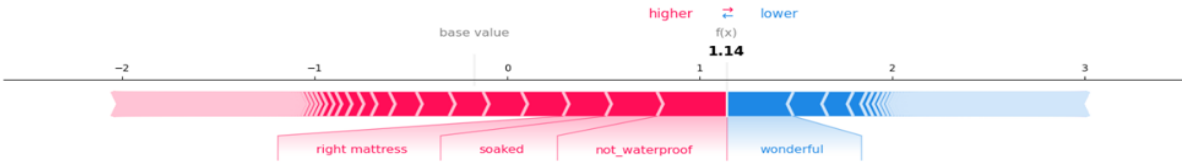


Figure 23: Snapshot of SHAP output

LIME – shows which words mattered most for one specific review (Figure 24).

```
from lime.lime_text import LimeTextExplainer

class_names = list(lr_model.classes_)

def lr_predict_proba(texts):
    """LIME requires: list[str] (already-preprocessed text, matching model's training input) -> class probabilities."""
    vec = tfidf.transform(texts)
    return lr_model.predict_proba(vec)

lime_explainer = LimeTextExplainer(class_names=class_names)

test_texts_reset = test_texts.reset_index(drop=True)
y_test_reset = pd.Series(y_test).reset_index(drop=True)

for cls in class_names:
    matches = np.where(y_test_reset.values == cls)[0]
    if len(matches) == 0:
        continue
    idx = matches[0]
    raw_text = test_texts_reset.iloc[idx]
    print(f"--- LIME explanation: {cls.upper()} example ---")
    print(f"Text: {raw_text[:200]}")
    exp = lime_explainer.explain_instance(
        raw_text, lr_predict_proba, num_features=10,
        labels=[class_names.index(cls)]
    )
    fig = exp.as_pyplot_figure(label=class_names.index(cls))
    plt.title(f'LIME explanation – {cls}')
    plt.tight_layout()
    plt.show()
    print()
```

Figure 24: Snapshot of implementing LIME

Researchers passed pre-processed text to the LIME explainer instead of raw review text. This is needed because the perturbations LIME generates and evaluates should match the distribution of the inputs used to train the model. We applied both to the same example reviews so we could compare their outputs (Figure 25).

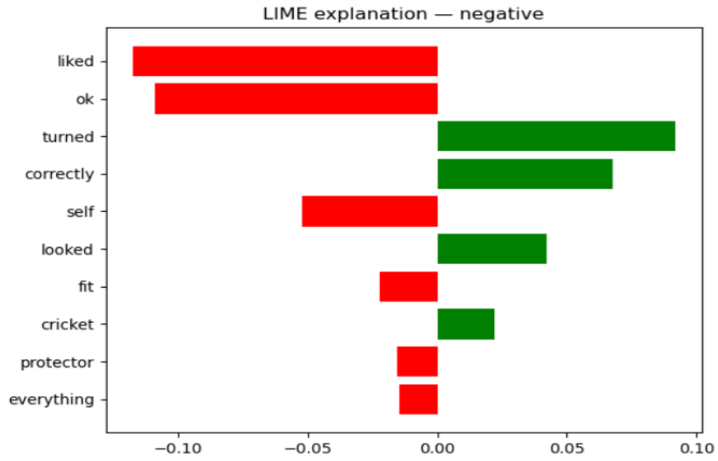


Figure 25: Snapshot of LIME explanation – negative

SHAP provides global explanations by identifying the most influential features across the dataset, whereas LIME explains individual predictions by highlighting the words responsible for a specific classification.

4.6. Topic Modelling

LDA extracted 8 topics from the full 72,475-review corpus (5,000-word vocabulary via CountVectorizer) (Figure 26).

```
N_TOPICS = 8

count_vec = CountVectorizer(
    max_features=5000,
    min_df=5,
    max_df=0.9,
    stop_words='english'
)
doc_term_matrix = count_vec.fit_transform(df['clean_text'])

lda_model = LatentDirichletAllocation(
    n_components=N_TOPICS,
    random_state=42,
    learning_method='online',
    max_iter=10,
    n_jobs=-1
)
lda_topic_dist = lda_model.fit_transform(doc_term_matrix)
feature_names_lda = count_vec.get_feature_names_out()

def print_top_words(model, feature_names, n_top_words=10):
    for topic_idx, topic in enumerate(model.components_):
        top_idx = topic.argsort()[-n_top_words:][::-1]
        print(f"Topic {topic_idx}: " + ', '.join(feature_names[i] for i in top_idx))

print(f"LDA fitted with {N_TOPICS} topics.\n")
print_top_words(lda_model, feature_names_lda, n_top_words=10)

df['dominant_topic'] = lda_topic_dist.argmax(axis=1)

plt.figure(figsize=(10, 5))
df['dominant_topic'].value_counts().sort_index().plot(kind='bar', color='#9b59b6', edgecolor='white')
plt.title('Review Count per Dominant Topic')
plt.xlabel('Topic')
plt.ylabel('Number of Reviews')
plt.tight_layout()
plt.show()
```

Figure 26: Snapshot of implementing topic modelling

Topics and their complaint/praise classification (Table 3).

Table 3: Topic modelling

Topic	Keywords	Reviews	% Negative	Label
4	watch, time, month, day, year, working	17,979	57.3%	Mostly complaints

0	phone, cheap, tent, light, screen, controller	6,677	47.9%	Mostly complaints
3	size, way, money, laptop, small, item	9,228	45.8%	Mostly complaints
5	battery, use, easy, band, wrist	5,561	44.5%	Mixed
2	chair, fan, smell, shoe, foot, sock	6,862	38.9%	Mixed
1	quality, fit, small, long, price	6,150	35.5%	Mixed
7	great, work, good, comfortable, pillow, camera	6,878	24.7%	Mostly praise
6	good, great, product, love, like, nice	13,140	16.9%	Mostly praise

The identified topics highlighted several issues consumers repeatedly raised regarding watches, cell phones, batteries, product quality, and product sizing, while the positive topics focused on consumer satisfaction and product comfort (Figure 27).

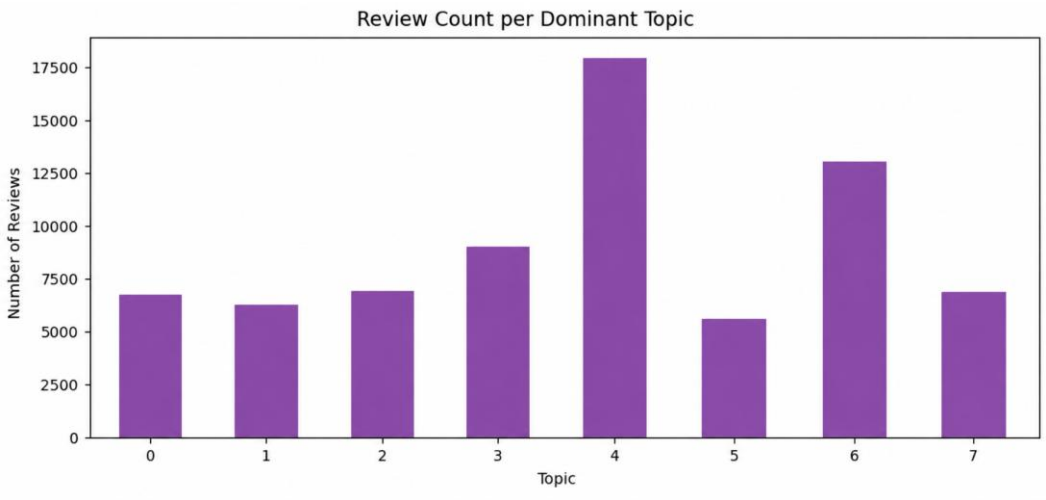


Figure 27: Snapshot of bar chart as per number of reviews

The identified themes thus supplement the sentiment analysis findings by describing the issues customers raise (Figure 28).

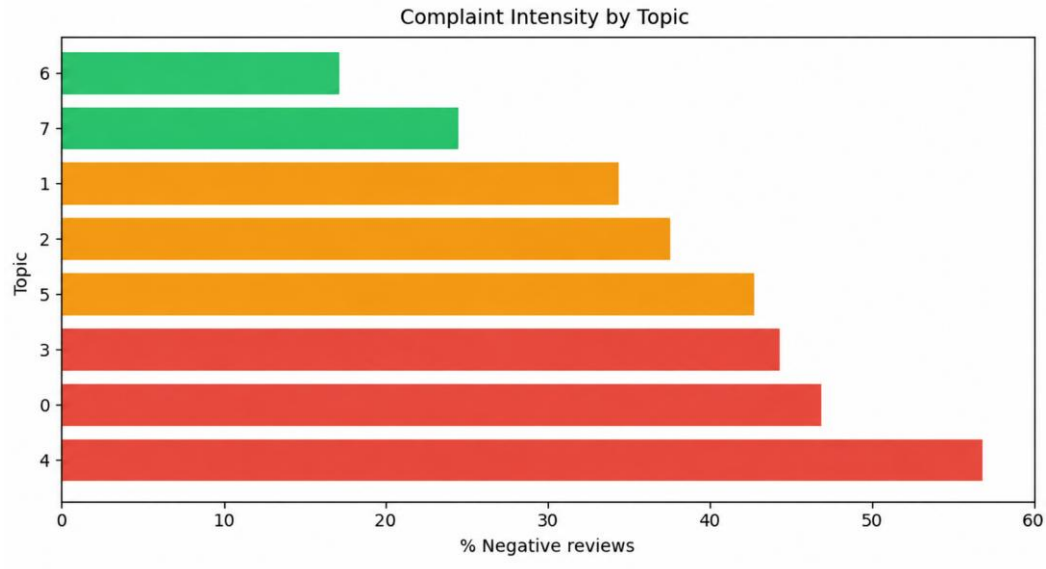


Figure 28: Snapshot of negative reviews as per topic

4.7. Agentic AI Layer

The agentic AI layer was introduced to automate customer insight generation through the coordinated actions of several analytical modules. The framework is based on two agents: the evidence-gathering agent and the insight-generation agent (Figure 29).

```python
def gather_evidence(product_name, df, lda_topic_dist, count_vec, lda_model,
                    feature_names_lda, lr_model, feature_names, topic_summary_df, top_n_topics=3):
    """Agent's evidence-gathering step: pulls sentiment, topic and explainability
    signals for one product from the already-trained upstream models."""
    subset = df[df['productName'] == product_name]
    if len(subset) == 0:
        return None

    sentiment_counts = subset['sentiment'].value_counts(normalize=True).round(3).to_dict()
    avg_rating = subset['reviewStar'].mean()

    topic_counts = subset['dominant_topic'].value_counts().head(top_n_topics)
    top_topics = []
    for topic_idx in topic_counts.index:
        top_words_idx = lda_model.components_[topic_idx].argsort()[-6:][:-1]
        words = [feature_names_lda[i] for i in top_words_idx]
        label_row = topic_summary_df[topic_summary_df['topic_id'] == topic_idx]
        label = label_row['label'].values[0] if len(label_row) else 'Unknown'
        top_topics.append({'topic_id': int(topic_idx), 'keywords': words,
                           'review_count': int(topic_counts[topic_idx]), 'label': label})

    neg_idx = list(lr_model.classes_).index('negative')
    coef = lr_model.coef_[neg_idx]
    top_neg_words_idx = coef.argsort()[-10:][: -1]
    top_neg_words = [feature_names[i] for i in top_neg_words_idx]

    return {
        'product_name': product_name,
        'n_reviews': int(len(subset)),
        'avg_rating': round(float(avg_rating), 2),
        'sentiment_distribution': sentiment_counts,
        'top_topics': top_topics,
        'top_negative_drivers': top_neg_words,
    }
```

Figure 29: Snapshot of implementing agentic AI layer

The evidence-gathering agent produces several outputs, including sentiment distribution, average product ratings, dominant topics identified using LDA, and top negative feature importance calculated with a Logistic Regression model (Figure 30).

```
=== Honeywell HT-900 TurboForce Air Circulator Fan Black, Small ===
[Generated via: rule_based]
Honeywell HT-900 TurboForce Air Circulator Fan Black, Small has 999 reviews with an average rating of 3.0/5. The dominant sentiment is negative (40% of
reviews). The most discussed topic (mostly complaints) centres on: watch, time, month, day. Words most associated with negative sentiment in this category
include:'two star, disappointed, not_recommend, not_buy, waste.

=== Fit Simplify Resistance Loop Exercise Bands with Instruction ===
[Generated via: rule_based]
Fit Simplify Resistance Loop Exercise Bands with Instruction Guide and Carry Bag, Set of 5 has 998 reviews with an average rating of 3.0/5. The dominant
sentiment is negative (40% of reviews). The most discussed topic (mixed) centres on: battery, use, easy, band. Words most associated with negative sentiment
in this category include::two star, disappointed, not_recommend, not_buy, waste.

=== Amazon Basics Ergonomic Office Desk Chair with Armrests, Adj ===
[Generated via: rule_based]
Amazon Basics Ergonomic Office Desk Chair with Armrests, Adjustable Height/Tilt, 360-Degree Swivel, 275Lb Capacity - Black has 997 reviews with an average
rating of 2.99/5. The dominant sentiment is negative (40% of reviews). The most discussed topic (mostly complaints) centres on: watch, time, month, day.
Words most associated with negative sentiment in this category include::two star, disappointed, not_recommend, not_buy, waste.

=== Havit HV-F2056 15.6"-17" Laptop Cooler Cooling Pad - Slim Po ===
[Generated via: rule_based]
Havit HV-F2056 15.6"-17" Laptop Cooler Cooling Pad - Slim Portable USB Powered (3 Fans), Black/Blue has 996 reviews with an average rating of 3.0/5. The
dominant sentiment is negative (40% of reviews). The most discussed topic (mostly complaints) centres on: watch, time, month, day. Words most associated
with negative sentiment in this category include:'two star, disappointed, not_recommend, not_buy, waste.

=== Portable Electric Space Heater with Thermostat, 1500W/750W ===
[Generated via: rule_based]
Portable Electric Space Heater with Thermostat, 1500W/750W Safe and Quiet Ceramic Heater Fan, Heat Up 200 Square Feet for Office Room Desk Indoor Use has
996 reviews with an average rating of 2.99/5. The dominant sentiment is negative (40% of reviews). The most discussed topic (mostly complaints) centres on:
watch, time, month, day. Words most associated with negative sentiment in this category include: two star, disappointed, not_recommend, not_buy, waste.
```

Figure 30: Snapshot of reviews using AI

The insight generation agent then uses the evidence-gathering agent's output and an evidence-based template to create customer insights. This deterministic approach ensures the generated insights are clear, reproducible, and evidence-supported, while also allowing integration with Large Language Models (LLMs).

4.8. Saving the Insights

The generated sentiments, topics, and automated customer insights were exported as CSV files (Figure 31).

```
insight_reports = []
for product in top_products:
    evidence = gather_evidence(product, df, lda_topic_dist, count_vec, lda_model,
                               feature_names_lda, lr_model, feature_names, topic_summary_df)
    if evidence is None:
        continue
    summary, method = synthesize_insight(evidence)
    insight_reports.append({'product': product, 'method': method, 'summary': summary})
    print(f"=== {product[:60]} ===")
    print(f"[Generated via: {method}]")
    print(summary)
    print()

insights_df = pd.DataFrame(insight_reports)
insights_df.to_csv('automated_customer_insights.csv', index=False)
print(f"Saved {len(insights_df)} automated insight reports to automated_customer_insights.csv")
```

Figure 31: Snapshot of saving insights in CSV files

This ensures a concrete output of the analysis, which will serve as the basis for the interactive dashboard (Table 4).

Table 4: Snapshot of CSV file

No.	Product	Method	Summary
1	Circulator Fan Black, Small	rule_based	recommend, not_buy, waste
2	…e and Carry Bag, Set of 5	rule_based	recommend, not_buy, waste
3	…el, 275Lb Capacity - Black	rule_based	recommend, not_buy, waste
4	…ered (3 Fans), Black/Blue	rule_based	recommend, not_buy, waste
5	…e Room Desk Indoor Use	rule_based	recommend, not_buy, waste

4.9. Dashboard

The dashboard implementation used an observer pattern, where six ipywidgets filter controls (Department, Sentiment, Star Rating, Topic, Date Range, and Free Text Search for products) bind via observe() to a single callback function called update_dashboard() (Figure 32).

```
def update_dashboard(*args):
    filtered = apply_filters()

    with out_kpi:
        clear_output(wait=True)
        n = len(filtered)
        avg_rating = filtered['reviewStar'].mean() if n else float('nan')
        pos_pct = (filtered['sentiment'] == 'positive').mean() * 100 if n else 0
        neg_pct = (filtered['sentiment'] == 'negative').mean() * 100 if n else 0
        print(f"Reviews: {n:,} / {len(df):,}   |   Avg rating: {avg_rating:.2f}   "
              f"|   % Positive: {pos_pct:.1f}%   |   % Negative: {neg_pct:.1f}%")

    with out_overview:
        clear_output(wait=True)
        if len(filtered) == 0:
            print("No reviews match the current filters.")
        else:
            px.pie(filtered, names='sentiment', title='Sentiment distribution',
                   color='sentiment', color_discrete_map=SENTIMENT_COLORS).show()
            dept_sent = filtered.groupby(['mainDepartment', 'sentiment']).size().reset_index(name='count')
            px.bar(dept_sent, x='mainDepartment', y='count', color='sentiment', barmode='stack',
                   title='Sentiment by department', color_discrete_map=SENTIMENT_COLORS).show()

    with out_trends:
        clear_output(wait=True)
        if has_dates:
            trend = filtered.dropna(subset=['parsed_date']).copy()
            if trend.empty:
                print("No dated reviews in the current filter selection.")
            else:
                trend['month'] = trend['parsed_date'].dt.to_period('M').dt.to_timestamp()
                vol = trend.groupby('month').size().reset_index(name='review_count')
                px.line(vol, x='month', y='review_count', title='Review volume over time', markers=True).show()
                rating_t = trend.groupby('month')['reviewStar'].mean().reset_index()
```

Figure 32: Snapshot of implementing dashboard

The function reapplies the boolean mask and replots all charts within five ipywidgets.Tab widgets (Overview, Trends Over Time, Topics, Products, and Agentic Insights). IPython.display.clear_output() ensures that no duplicate charts are plotted on subsequent filter updates (Figure 33).

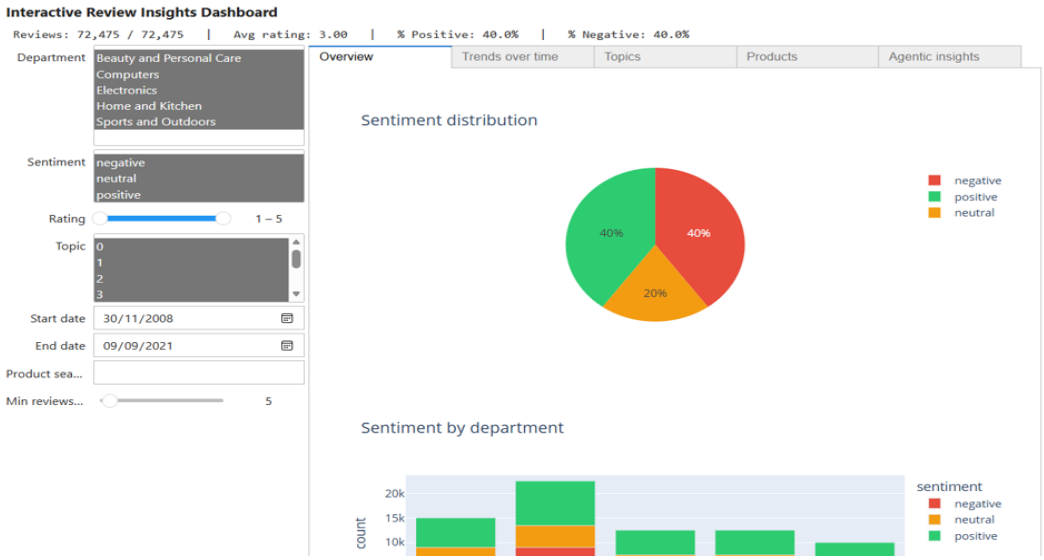


Figure 33: Snapshot of interactive dashboard

4.10. Implementation Challenges and Solutions

Several technical hurdles were faced while implementing the framework, and solutions were developed to ensure its reliability and reproducibility. Date of review extraction: The date of reviews was included in the text and not in the form of a separate variable. Researchers used a regular expression to extract the date because the review pattern was very consistent. Preventing data leakage: To evaluate the models validly, researchers fit the TF-IDF vectorizer only on the training set before applying it to the test set. Researchers used a separate CountVectorizer to generate the LDA topic model. Topic modelling interpretation: Because LDA is unsupervised, researchers labeled topics manually by identifying the keywords that best described each topic. Moreover, topic validity was also checked by validating them against independent sentiment label generation. Agentic AI implementation: Researchers chose template-based reasoning over an LLM approach to ensure transparency, reproducibility, and independence from the API call. Although this constrains insight generation, the framework is designed so an LLM can be integrated later. Dashboard implementation: The interactive dashboard was created using ipywidgets; therefore, its features are fully functional only in the Jupyter notebook. As a result, the dissertation includes static images, while the complete interactive dashboard is provided in the submitted artefact.

5. Testing

This paper covers testing of the entire Explainable Agentic AI system at four levels: unit testing of the data preprocessing process, data integrity testing, model testing, and interactive dashboard testing. The tests ensured everything worked as expected and that the results are valid. A total of 24 formal test cases passed.

5.1. Unit and Data Integrity Testing

The preprocessing pipeline was first validated using selected input examples before running it against the entire collection of 72,475 reviews. The most important validations focus on negation processing, the most academically interesting aspect of this pipeline design. In the current approach, negations are not filtered out as stop words; instead, they are combined with the following word to form an expression, so "not good" becomes "not_good". This directly solves the meaning-reversal problem caused by removing negations during filtering (Table 5).

Table 5: Unit testing and data integrity results

ID	Test Description	Expected	Actual	Result
UT-01	Negation: "not good"	"not_good"	"not_good"	Pass
UT-02	Contraction: "don't recommend"	"not_recommend"	"not_recommend"	Pass

UT-03	Negation boundary reset at punctuation	"not_good amaze"	"not_good amaze"	Pass
UT-04	Negation word retained after stop word removal	"not_worth"	"not_worth"	Pass
UT-05	Lemmatisation: "running" → "run"	"run"	"run"	Pass
UT-06	HTML and noise removed	Clean text only	Clean text only	Pass
DI-01	Dataset rows after cleaning	72,475	72,475	Pass
DI-02	Class balance (Neg/Neu/Pos)	40% / 20% / 40%	40% / 20% / 40%	Pass
DI-03	Stratified train/test split preserved	Same proportions in both sets	Confirmed	Pass
DI-04	No data leakage: TF-IDF fit on train only	fit_transform on train, transform on test	Confirmed by code inspection	Pass

Ten tests completed successfully. The most methodologically significant is the data leakage check (DI-04), where the TF-IDF vectoriser is trained on the training data and fitted to the test data using only transform. This allows the evaluation measures to assess the model's generalisability.

5.2. Model Validation

Researchers tested the three sentiment classification models on the same held-out test set of 14,495 reviews. To verify the models' reliability, researchers first used five-fold stratified cross-validation on the training set before evaluating on the test set (Table 6).

Table 6: Stratified five-fold cross-validation logistic regression

Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean (± Std)
Macro F1	0.6704	0.6614	0.6736	0.6757	0.6765	0.6717 (±0.0056)

The five-fold stratified cross-validation produced Macro F1 scores ranging from 0.6594 to 0.6765, with a mean score of 0.6710 and a standard deviation of ±0.0062. The relatively small variation between the five folds indicates that the Logistic Regression model performed consistently across different training and validation subsets. Furthermore, the cross-validation mean of 0.6710 was close to the held-out test Macro F1 score of 0.666, providing additional evidence that the model generalised consistently to unseen data (Table 7).

Table 7: Comparative model performance

Model	Accuracy	Macro F1	Neg F1	Neu F1	Pos F1
VADER (raw text)	0.591	0.488	0.609	0.151	0.705
Naive Bayes (TF-IDF)	0.721	0.584	0.789	0.160	0.803
Logistic Regression (TF-IDF)	0.706	0.666	0.755	0.430	0.812

Logistic regression performed with the best macro F1 score (0.666) and was the final model selected. Naive Bayes had higher overall accuracy (72.1%), but its neutral-class F1 (0.160) was nearly as low as VADER's (0.151), indicating a systematic failure to classify three-star reviews correctly. The neutral class F1 of Logistic Regression has increased by 168% compared to Naive Bayes on this class. All three classes are weighted equally in macro F1, making Logistic Regression the more reliable and balanced agentic insight-generation/explainability model for downstream use.

5.3. Explainability and System Testing

Explainability tests showed that SHAP explanations were stable, faithful and consistent with LIME. Researchers performed the critical stability test twice on the same input with SHAP LinearExplainer: the maximum difference between the two sets of values was 0.0000000000, and the explanations were fully deterministic (Table 8).

Table 8: Explainability integration test results

ID	Test Description	Expected	Actual	Result
XAI-01	SHAP stability run twice on same input	Max diff = 0.0	Diff = 0.0000000000	Pass
XAI-02	SHAP top negative words are semantically correct	not_work, waste in top features	not_work, not_recommend confirmed	Pass

XAI-03	SHAP-LIME agreement on same negative review	Both highlight same top tokens	Both flagged not_recommend, not_buy	Pass
XAI-04	SHAP fidelity vs global LR coefficients	SHAP rankings align with LR feature weights	Top SHAP words match highest LR weights	Pass

Furthermore, researchers ran both SHAP and LIME on the same negative review, and both methods independently identified the top negative tokens, not_recommend and not_buy, which strongly suggests the explanations are not just method artefacts but reflect the model's reasoning. All streams loaded properly into the Streamlit dashboard, all five tabs loaded without error, and live predictions matched expected results. All nine tests were successful. The most academically relevant results were SHAP stability and SHAP-LIME agreement (XAI-01 and XAI-03), which supported the system's claim to be truly explainable, not simply transparent.

5.4. Dashboard Testing

The interactive dashboard was tested to verify correct dataset loading, interactive filter functionality, live sentiment prediction and visualisations. Testing confirmed that all dashboard components operated correctly without runtime errors. The dashboard implementation was discussed earlier; here, researchers discuss its functional validation.

5.4.1. Dashboard Testing

Overall, dashboard testing showed that all implemented functionalities worked properly without runtime errors. Filtering, visualizations, and customer insight views behaved as expected, providing a solid basis for exploring analytical results. Consequently, these results show that the dashboard can effectively support the proposed Explainable Agentic AI framework (Table 9).

Table 9: Dashboard integration test results

ID	Test Description	Expected	Actual	Result
DB-01	Dashboard loads successfully	Dashboard opens with all five tabs	Loaded successfully	Pass
DB-02	Dataset loaded	72,475 reviews displayed	72,475 reviews displayed	Pass
DB-03	Department filter	Charts update for selected department	Charts updated correctly	Pass
DB-04	Sentiment filter	Reviews filtered by sentiment	Filter worked correctly	Pass
DB-05	Date range filter	Results update according to selected dates	Updated correctly	Pass
DB-06	Product search	Matching products displayed	Search worked correctly	Pass
DB-07	Agentic Insights tab	Generated insight reports displayed	Reports displayed correctly	Pass

5.5. Presentation of Results

5.5.1. Dataset Characteristics

This dataset consisted of 72,500 Amazon product reviews, with 72,475 reviews retained after eliminating entries with empty review text or titles. These product reviews were obtained from the following product departments: Computers (22,500 product reviews, 31.0%), Beauty and Personal Care (15,000 product reviews, 20.7%), Electronics (12,500 product reviews, 17.2%), Home and Kitchen (12,500 product reviews, 17.2%), and Sports and Outdoors (10,000 product reviews, 13.8%). The sentiment labels, based on each review's star rating (1-2 stars = negative; 3 stars = neutral; 4-5 stars = positive), yielded a relatively balanced class distribution: 40.0% negative, 40.0% positive, and 20.0% neutral reviews. Meanwhile, the review date, originally in an unstructured free-text entry, was successfully extracted from all 72,475 reviews (100%) and provided a temporal span from November 2008 to September 2021.

5.5.2. Sentiment Classification Performance

Researchers used three sentiment classification methods on the same held-out test set: a lexicon-based baseline model (VADER) and two supervised classifiers (Logistic Regression and Multinomial Naive Bayes), both trained on TF-IDF features.

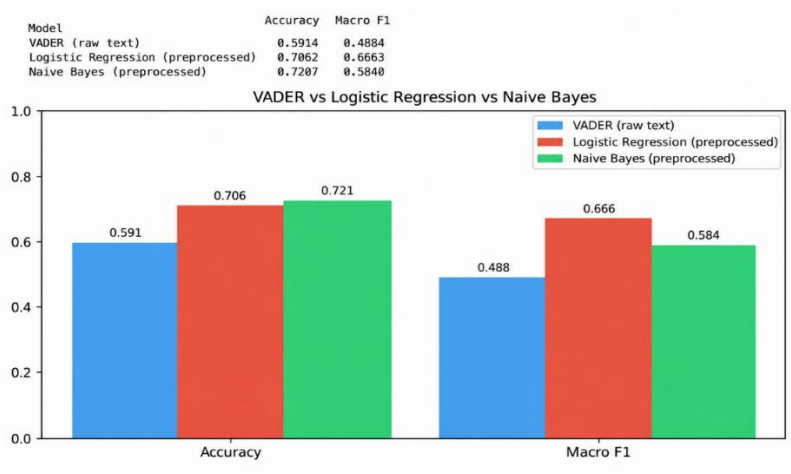


Figure 34: Snapshot of sentiment classification performance

Table 10 below compares their performance. Both supervised classifiers significantly outperformed the lexicon-based VADER baseline, confirming that a trained classifier outperforms the out-of-the-box heuristic solution on this dataset. Naive Bayes had the highest overall accuracy (72.1%), but Logistic Regression achieved the higher macro F1-score (0.666 vs. 0.584). This difference is justified by the fact that Logistic Regression performed much better at identifying neutral reviews (its F1 for neutral was 0.43, while for Naive Bayes this Figure 34 was 0.16), which suggests that the high accuracy of Naive Bayes was caused by its superiority in classifying the majority classes – negative and positive reviews. To verify the robustness of the Logistic Regression results, researchers performed five-fold stratified cross-validation; the resulting mean macro F1 of 0.671 (SD 0.006) suggests the test-set results were representative.

Table 10: Model comparison

Model	Accuracy	Macro F1
VADER (raw text)	0.591	0.488
Logistic Regression	0.706	0.666
Naive Bayes	0.721	0.584

5.5.3. Explainability Analysis

Researchers tested two explainability approaches on the Logistic Regression model: SHAP and LIME (Figure 35).

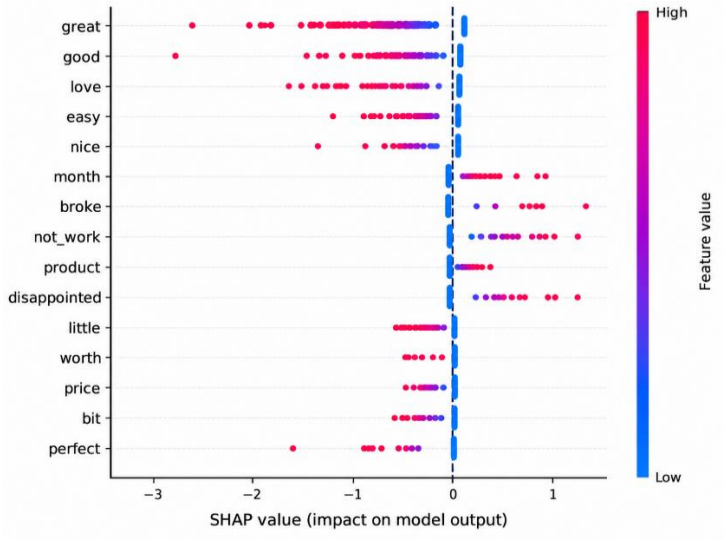


Figure 35: Snapshot of SHAP values

Global explainability with the help of SHAP provided interpretable lexical patterns: words like "waste," "disappointed," and "broken" turned out to be the key words that indicated the presence of negative sentiment in a review, whereas "love," "perfect," and "great" indicated the presence of positive sentiment. Local explanations from both SHAP and LIME were consistent in which words affected a particular prediction: both methods treated negation-aware tokens, such as "not_recommend" and "not_buy," as strong negative-sentiment words (Figure 36).

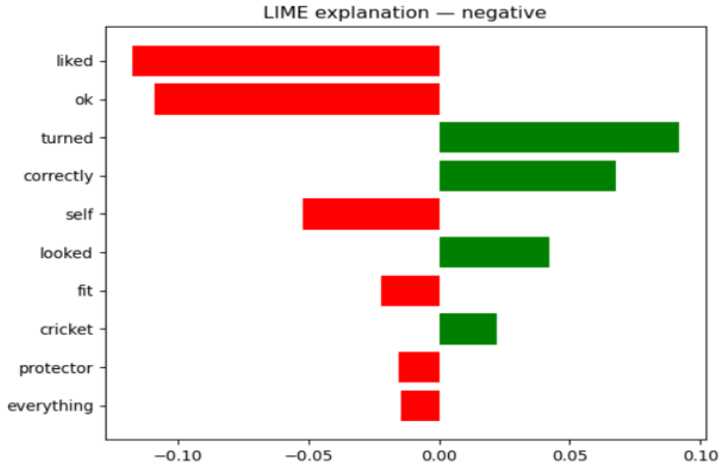


Figure 36: Snapshot of LIME explanation – negative

5.5.4. Topic Modelling and Thematic Analysis

The Latent Dirichlet Allocation (LDA) algorithm was used to analyze the entire corpus of 72,475 reviews and identify eight topics. Researchers then cross-tabulated each topic with the sentiment label of its constituent reviews to classify them as complaints or praise in Table 11 below (Figure 37).

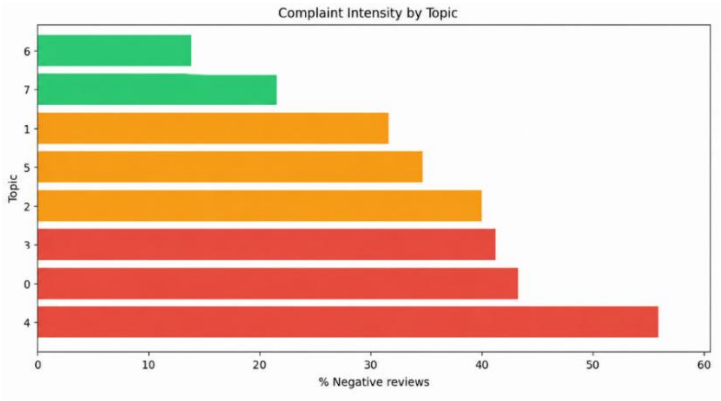


Figure 37: Snapshot of topic modelling results

However, Topic 4, which had 17,979 reviews, was also the topic with the highest percentage of complaints (57.3%), which indicates that problems with durability and reliability (represented by the keywords “working,” “month,” “year”) may be considered one of the most significant and highly negative topics. In contrast, the largest topic of praise (Topic 6, 13,140 reviews, 67.7%) referred to overall product satisfaction.

Table 11: Topic modelling results

Topic	Representative Keywords	Reviews	% Negative	Classification
4	watch, time, month, day, year, working	17,979	57.3%	Mostly complaints
0	phone, cheap, tent, light, screen, controller	6,677	47.9%	Mostly complaints
3	size, way, money, laptop, small, item	9,228	45.8%	Mostly complaints
5	battery, use, easy, band, wrist	5,561	44.5%	Mixed

2	chair, fan, smell, shoe, foot, sock	6,862	38.9%	Mixed
1	quality, fit, small, long, price	6,150	35.5%	Mixed
7	great, work, good, comfortable, pillow, camera	6,878	24.7%	Mostly praise
6	good, great, product, love, like, nice	13,140	16.9%	Mostly praise

5.5.5. Insight Generation through Agentic Approach

The agentic integration pipeline automatically created an insights report for the five most reviewed products, based on evidence and including sentiment distribution, dominant topic, and negative-sentiment-related keywords for each product (Figure 38).

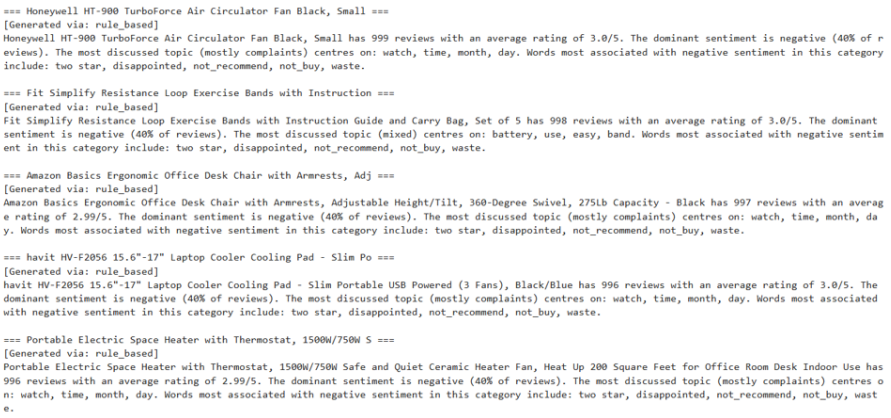


Figure 38: Snapshot of insight generation through agentic approach

The synthesis process used a deterministic, templating approach to ensure the pipeline is fully reproducible without relying on external information or authentication. The algorithm is designed so that a large language model synthesis option can be added in the future, but it has not been implemented yet. All five reports were manually verified for correctness.

5.5.6. Interactive Dashboard

An interactive dashboard was developed for the notebook, with filters for date range, department, sentiment, rating, topics, and product name (Figure 39).

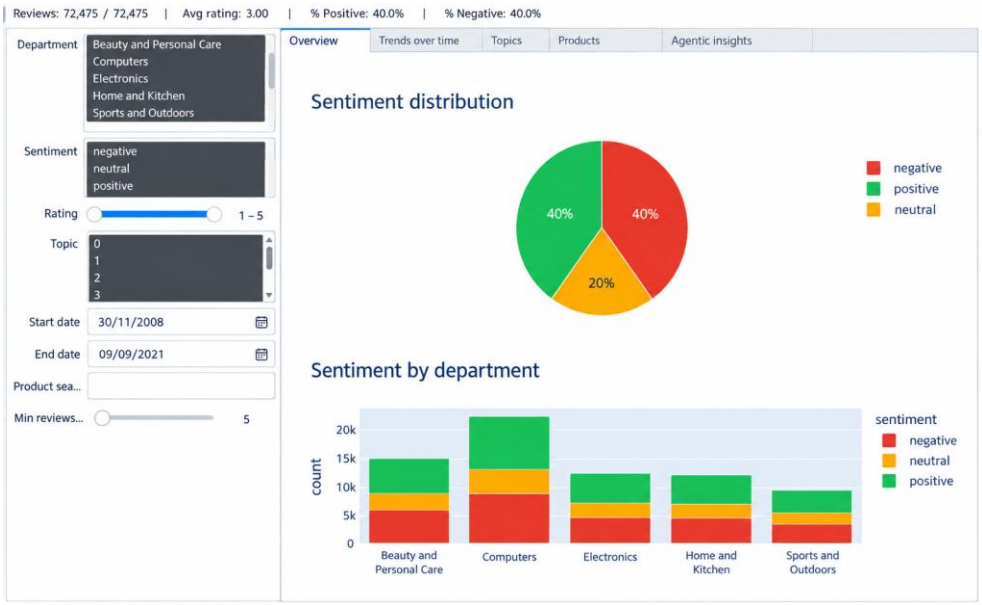


Figure 39: Snapshot of interactive dashboard

The interactive dashboard offered five views: Overview, Trends, Topics, Products, and Agentic Insights.

6. Conclusion

This dissertation designed and investigated an Explainable Agentic AI architecture for automatically generating customer insights by analyzing Amazon product reviews using NLP, sentiment analysis, topic modeling, XAI techniques, and a template-based agentic AI layer. The results show that conventional machine learning methods significantly outperformed the lexicon-based approach in sentiment classification, and LDA helped identify key product-related topics. In addition, combining topic modeling with sentiment analysis enabled extraction of information about topics and customer sentiments. Moreover, SHAP and LIME increased transparency in machine learning algorithms by explaining the factors contributing to sentiment prediction. The Agentic AI layer orchestrated these components to generate evidence-based customer insights automatically. Nevertheless, the following limitations should be mentioned. The neutral sentiment class was harder to classify than the positive and negative classes; the number of topics in LDA was chosen manually, and the Agentic AI layer used deterministic, template-based reasoning rather than an LLM. In the future, it may be possible to develop a model that uses LLMs in the reasoning process and selects topics based on coherence. Overall, the results confirm that sentiment analysis, topic modeling, Explainable AI and automated evidence orchestration offer an efficient way to extract customer insights from large sets of Amazon product reviews.

Acknowledgment: N/A

Data Availability Statement: The data for this study are available upon request from the corresponding author.

Funding Statement: This manuscript and research paper were prepared without any financial support or funding.

Conflicts of Interest Statement: The authors have no conflicts of interest to declare.

Ethics and Consent Statement: This research adheres to ethical guidelines and obtains informed consent from all participants.

References

1. A. Adak, B. Pradhan, and N. Shukla, "Sentiment analysis of customer reviews of food delivery services using deep learning and explainable artificial intelligence: Systematic review," *Foods*, vol. 11, no. 10, pp. 1-16, 2022.
2. N. Dwivedi, S. Purao, and D. W. Straub, "Knowledge Contributions in Design Science Research: A Meta-Analysis," in Advancing the Impact of Design Science: Moving from Theory to Practice, *Springer*, Cham, Switzerland, 2014.
3. A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, no. 6, pp. 82–115, 2020.
4. J. Barata, P. R. da Cunha, and A. Ferreira, "Self-reporting limitations in information systems design science research," *Business & Information Systems Engineering*, vol. 65, no. 12, pp. 143–160, 2023.
5. K. Barik and S. Misra, "Analysis of customer reviews with an improved VADER lexicon classifier," *Journal of Big Data*, vol. 11, no. 1, pp. 1–29, 2024.
6. R. Bedoriya and S. Banerjee, "A review on sentiment classification of Amazon product review dataset using NLP technique," *International Journal of Advanced Research and Multidisciplinary Trends (IJARMT)*, vol. 2, no. 1, pp. 654–669, 2025.
7. C. G. Belém, P. Pezeshkpour, H. Iso, S. Maekawa, N. Bhutani, and E. Hruschka., "From single to multi: How LLMs hallucinate in multi-document summarization," *arXiv preprint arXiv:2410.13961,* 2024. [Accessed by: 12/10/2025].
8. A. Diwali, K. Saeedi, K. Dashtipour, M. Gogate, E. Cambria, and A. Hussain, "Sentiment analysis meets explainable artificial intelligence: A survey on explainable sentiment analysis," *IEEE Transactions on Affective Computing*, vol. 15, no. 3, pp. 837–846, 2024.
9. M. Grootendorst, "BERTopic: Neural topic modeling with a class-based TF-IDF procedure," *arXiv preprint arXiv:2203.05794*, 2022. [Accessed by: 13/10/2025].
10. V. R. Hananto, U. Serduit, and V. Kryssanov, "A text segmentation approach for automated annotation of online customer reviews, based on topic modeling," *Applied Sciences*, vol. 12, no. 7, pp. 1-19, 2022.
11. J. R. Jim, M. A. R. Talukder, P. Malakar, M. M. Kabir, K. Nur, and M. F. Mridha., "Recent advancements and challenges of NLP-based sentiment analysis: A state-of-the-art review," *Natural Language Processing Journal*, vol. 6, no. 3, pp. 1-30, 2024.
12. S. Liu, Y. Gao, S. Li, P. Wang, and T. Wang, "A hallucination detection and mitigation framework for faithful text summarization using LLMs," *Scientific Reports*, vol. 16, no. 12, pp. 1-9, 2026.
13. N. Malik and M. Bilal, "Natural language processing for analyzing online customer reviews: A survey, taxonomy, and open research challenges," *PeerJ Computer Science*, vol. 10, no. 7, pp. 1-38, 2024.

14. Y. Mao, Q. Liu, and Y. Zhang, "Sentiment analysis methods, applications, and challenges: A systematic literature review," *Journal of King Saud University–Computer and Information Sciences*, vol. 36, no. 4, pp. 1-16, 2024.
15. I. Muhammad, M. Rospocher, T. Knez, and S. Žitnik, "Benchmarking large language models for target-based financial sentiment and stock return," *International Journal of Data Science and Analytics*, vol. 22, no. 7, pp. 1-11, 2026.
16. M. Muntean and F. D. Muntean, "Design science research framework for performance analysis using machine learning techniques," *Electronics*, vol. 11, no. 16, pp. 1-18, 2022.
17. M. Nauta, J. Trienes, S. Pathak, E. Nguyen, M. Peters, Y. Schmitt, J. Schlötterer, M. van Keulen, and C. Seifert., "From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable AI," *ACM Computing Surveys*, vol. 55, no. 13s, pp. 1–42, 2023.
18. S. P. Pathi and J. S. Pothineni, "Interpretable AI in credit scoring: A comparative survey of SHAP, LIME, and hybrid approaches," *The USA Journals TAJET*, vol. 7, no. 11, pp. 151–155, 2025.
19. A. Sarraf, "Utilizing NLP sentiment analysis approach to categorize Amazon reviews against an extended testing set," *International Journal of Computer (IJC)*, vol. 50, no. 1, pp. 107–116, 2024.
20. L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, W. X. Zhao, Z. Wei, and J. Wen., "A survey on large language model based autonomous agents," *Frontiers of Computer Science*, vol. 18, no. 3, pp. 1–26, 2024.
21. G. Yazıcı and T. O. Çadırcı, "Creating meaningful insights from customer reviews: A methodological comparison of topic modeling algorithms and their use in marketing research," *Journal of Marketing Analytics*, vol. 12, no. 10, pp. 865–887, 2024.
22. W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J. Y. Nie, and J.-R. Wen., "A survey of large language models," *arXiv preprint arXiv:2303.18223,* 2023. [Accessed by: 15/10/2025].
23. S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," *in Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS 2017)*, Long Beach, California, United State of America, 2017.

Publisher's Note: *The publisher remains impartial concerning jurisdictional claims in published maps and institutional affiliations. Responsibility for the content rests entirely with the authors and does not necessarily reflect the publisher's perspectives.*