CCP - Report

Sentiment Analysis For Product Overview

Artificial Intelligence & Expert Systems

Course Code: CT-361

Made By:

Wajih Ur Rehman	CT-22083
Filza Tanveer	CT-22057
Taqi Haider	CT-22092

Submitted To: Sir Abdullah Siddiqui

1. INTRODUCTION

1.1. ABSTRACT

This project focuses on performing sentiment analysis on product reviews, aiming to classify customer feedback as positive or negative. By analyzing user-generated content, the system helps businesses gain insights into customer satisfaction and product performance. The project involves data preprocessing, text processing, feature extraction using TF-IDF, and training multiple machine learning models: Logistic Regression and Support Vector Machine (SVM). The evaluation of these models is performed using metrics such as accuracy and classification reports, providing a detailed understanding of each model's performance. This analysis can assist companies in better understanding consumer needs and improving their products accordingly.

1.2. AIMS AND OBJECTIVES

- To develop a sentiment analysis model for classifying product reviews as positive or negative.
- To preprocess raw text data by removing noise, stop words, and applying lemmatization.
- To implement feature extraction using TF-IDF for converting text data into numerical format.
- To train and compare the performance of two machine learning models: Logistic Regression and SVM.
- To evaluate the performance of the models using metrics such as accuracy and classification reports.

1.3. METHODOLOGY

- **Data Collection:** The dataset consists of product reviews gathered from an online source.
- **Data Preprocessing:** The text data is cleaned by removing special characters, stop words, and applying lemmatization.
- **Feature Extraction:** TF-IDF vectorization is used to convert the preprocessed text into numerical features suitable for machine learning.
- **Model Training:** Two different machine learning models (Logistic Regression, SVM) are trained on the feature-extracted data.
- **Evaluation:** The models' performances are assessed using metrics such as accuracy and classification reports to identify the most effective classifier.
- **Deployment:** The final selected model can be used for real-time sentiment analysis of new product reviews. We used streamlit for developing the application for deployment

1.4. TERMINOLOGIES

Stop Words

- **Code***:* `stop_words = set(stopwords.words('english'))`
- **Purpose**: Common words (e.g., the, is) that carry little meaning are excluded.
- **Justification**: Reduces noise and improves focus on impactful terms.

Tokenization

- ***Code****:* `tokens = word_tokenize(review)`
- ***Purpose****:* Splits text into individual words or tokens.
- ***Justification****:* Prepares text for filtering and lemmatization.

Stopword Removal

- ***Code:*** `[word for word in tokens if word.lower() not in stop_words]`
- ***Purpose:*** Removes non-informative words from tokens.
- ***Justification:*** Enhances model efficiency by focusing on relevant words.

Lemmatization

- ***Code:*** `[lemmatizer.lemmatize(token) for token in filtered_tokens]`
- ***Purpose:*** Converts words to their root form (e.g., *running* → *run*).
- ***Justification:*** Reduces variation in vocabulary and improves generalization.

Reconstruction

- ***Code:*** `' '.join(lemmatized_tokens)`
- ***Purpose:*** Joins tokens back into cleaned text.
- ***Justification:*** Required for input into vectorizers like TF-IDF.

Function Application

- ***Code:*** `df['cleaned_review'] = df['Review'].apply(preprocess_text)`
- ***Purpose:*** Applies preprocessing to each review.
- ***Justification:*** Ensures consistency across the dataset.

Dropping Original Text

- ***Code:*** `df = df.drop(['Review'], axis=1)`
- ***Purpose:*** Removes raw review column.
- ***Justification:*** Keeps dataset clean and model-ready.

WordCloud

- ***Code:*** **`WordCloud(...).generate(text)`**
- ***Definition:*** A visual representation of the most frequent words in the text.
- ***Justification:*** Helps in understanding dominant keywords in the reviews.

Sentiment Labeling

- *Code:* `def get_sentiment(rating): ...`
- ***Definition:*** Maps numerical scores to sentiment categories (positive, neutral, negative).
- ***Justification:*** Converts raw scores into meaningful categorical labels for classification.

Label Encoding

- *Code:* `LabelEncoder().fit_transform(...)`
- ***Definition:*** Converts sentiment labels into numerical format for model training.
- ***Justification:*** Machine learning models require numerical input.

Downsampling

- *Code:* `resample(..., n_samples=85000, ...)`
- ***Definition:*** Reduces the number of majority class samples to match minority classes.
- ***Justification:*** Addresses class imbalance to improve model performance and fairness.

Train-Test Split

- *Code:* `train_test_split(..., stratify=..., test_size=0.2)`
- ***Definition:*** Divides data into training and testing sets while preserving class distribution.
- ***Justification:*** Ensures fair evaluation of model performance.

TF-IDF Vectorization

- *Code:* `TfidfVectorizer(...).fit_transform(...)`
- ***Definition:*** Converts text into numerical features based on term importance.
- ***Justification:*** Prepares textual data for input into ML algorithms.

Pickle (Model/Data Serialization)

- *Code:* `pickle.dump(vect, file)`

- ***Definition:*** *Saves trained objects (like vectorizers) for reuse without retraining.*
- ***Justification:*** *Improves reproducibility and efficiency in deployment.*

GridSearchCV

- ***Code:*** `GridSearchCV(LogisticRegression(), param_grid, cv=5)`
- ***Definition:*** Performs an exhaustive search over specified hyperparameter values using cross-validation.
- ***Justification:*** Identifies the best-performing hyperparameters for the model to improve accuracy.

Hyperparameters (C, penalty)

- ***Code:*** `'C': [0.001, 0.01, ...], 'penalty': ['l1', 'l2']`
- ***Definition:*** `C` controls regularization strength, and `penalty` specifies the type of regularization (L1 or L2).
- ***Justification:*** Tuning these influences model complexity and prevents overfitting.

Logistic Regression

- ***Code:*** `LogisticRegression(penalty='l2', C=10, ...)`
- ***Definition:*** A linear model used for binary and multiclass classification.
- ***Justification:*** Effective and interpretable baseline model for classification tasks.

Support Vector Machine (SVM)

- ***Code:*** `SVC(kernel='linear')`
- ***Definition:*** A powerful classifier that finds the optimal hyperplane to separate classes.
- ***Justification:*** Performs well in high-dimensional spaces like TF-IDF text vectors.

Accuracy Score

- ***Code:*** `accuracy_score(y_test, y_pred)`
- ***Definition:*** Measures the proportion of correct predictions over total

predictions.

- ***Justification:*** Basic metric to evaluate model performance.

Classification Report

- ***Code:*** `classification_report(y_test, y_pred)`
- ***Definition:*** Provides precision, recall, and F1-score for each class.
- ***Justification:*** Offers deeper insights into class-wise performance beyond accuracy.

Confusion Matrix

- ***Code:*** `confusion_matrix(y_test, y_pred)`
- ***Definition:*** Tabular summary of actual vs. predicted classifications.
- ***Justification:*** Helps visualize model performance across classes.

Bar Plot for Model Comparison

- ***Code:*** `plt.bar(...)` and grouped bar chart logic
- ***Definition:*** A visual comparison of metrics like accuracy and loss across models.
- ***Justification:*** Simplifies interpretation and comparison of model outcomes.

Model Loss (assumed in df['Loss'])

- ***Definition:*** Quantifies the error between predicted and actual values (assumed precomputed).
- ***Justification:*** Evaluates model optimization and convergence alongside accuracy.

2. LITERATURE REVIEW

2.1. INTRODUCTION

In the digital age, the proliferation of online content has transformed the way individuals and organizations communicate, share opinions, and make decisions. Sentiment analysis, also known as opinion mining, has emerged as a critical field within natural language processing (NLP) that focuses on extracting and analyzing subjective information from textual data. By leveraging advanced computational techniques, sentiment analysis enables the identification and categorization of sentiments expressed in various forms of communication, such as social media posts, product reviews, and customer feedback. The three papers collectively explore the importance, challenges, and evolving methodologies of sentiment analysis in natural language processing, emphasizing its role in data analytics and the need for improved techniques to handle user-generated content

2.2. IMPORTANCE AND ROLE OF SENTIMENT ANALYSIS IN BUSINESS DECISION-MAKING

In the first paper [1], researchers have noted that data analytics is widely utilized across various industries and organizations to enhance business decision-making. They emphasize that by applying analytics to both structured and unstructured data, enterprises can significantly transform their planning and decision-making processes. In the authors' view, **sentiment analysis**, also referred to as opinion mining, plays a crucial role in everyday decision-making. These decisions can range from purchasing products (e.g., mobile phones) to reviewing movies and making investment choices—all of which have a substantial impact on daily life.

2.3. METHODOLOGIES IN SENTIMENT ANALYSIS

Overview of Common Techniques

The authors present a detailed survey of various methodologies and approaches to sentiment analysis, aiming to enhance understanding of the available techniques.

Machine Learning Algorithms in Sentiment Analysis

Researchers highlight that common machine learning algorithms, such as **Support Vector Machines (SVM)**, are frequently used in sentiment classification to distinguish between positive and negative sentiments.

Support Vector Machines (SVM)

Support Vector Machines are emphasized as a powerful supervised learning method commonly applied in sentiment classification. SVM works by finding the optimal hyperplane that best separates the data into sentiment classes (e.g., positive vs. negative). It's known for its high performance on small to medium-sized datasets and its effectiveness in handling high-dimensional feature spaces, such as those resulting from TF-IDF vectorization in text data.

2.4. CHALLENGES IN SENTIMENT ANALYSIS

- **Polarity Shift:**
  The authors observe that sentiment analysis encounters several challenges, including the change of sentiment polarity within a sentence or paragraph, which complicates classification.
- **Accuracy Issues:**
  Accuracy-related challenges such as misinterpretation of sarcasm, irony, and contextual ambiguity can hinder effective sentiment analysis.
- **Binary Classification Problems:**
  Sentiment classification is often reduced to binary classes (positive and negative), making it harder to capture nuanced sentiments like neutrality or mixed emotions.
- **Data Sparsity:**
  Sparse datasets—where most features (words) are zero or rarely used—can reduce model performance, particularly in text classification tasks.

2.5. THE IMPACT OF THE WORLD WIDE WEB ON SENTIMENT ANALYSIS

Data Generation through Social Media and E-commerce

In the second paper [2], researchers define sentiment analysis as the process of mining data, views, reviews, or sentences to predict the emotion conveyed in the text through **Natural Language Processing (NLP)**. In recent years, the **World Wide Web (WWW)** has emerged as a vast source of raw data generated by users.

Value of User-Generated Content

Researchers note that social media platforms and e-commerce websites like Facebook, Twitter, Amazon, and Flipkart enable users to freely share their views and feelings. This growing volume of user-generated content is recognized as a rich and valuable source of information for real-time business decision-making and market trend analysis.

2.6. ADVANCES AND INNOVATIONS IN SENTIMENT ANALYSIS TECHNIQUES

Recent Developments

Although sentiment analysis remains fundamentally text-based, the authors acknowledge the **challenges in accurately determining the polarity** of sentences. There's a pressing need to develop more advanced techniques to improve the reliability of sentiment detection.

Proposed New Techniques

In their paper [2], researchers present a survey of traditional and modern techniques used in sentiment analysis, along with **a new technique** proposed by the authors that aims to improve polarity detection.

2.7. EVOLUTION OF RESEARCH IN SENTIMENT ANALYSIS

Growth of Academic Interest

In the third paper [3], researchers identify sentiment analysis as a prominent research hotspot within **natural language processing (NLP)**. The field has received increasing attention from academic researchers over the past two decades.

Historical Trends and Emerging Hotspots

The authors observe a continuous rise in published research papers on sentiment analysis and note the lack of comprehensive survey works leveraging **keyword co-occurrence analysis**.

2.8. KEYWORD CO-OCCURRENCE ANALYSIS IN SENTIMENT RESEARCH

Methodology and Applications

Researchers present a study that focuses on surveying sentiment analysis trends using **keyword co-occurrence analysis** and **community detection algorithms**. These methods help identify relationships between different topics and techniques in sentiment analysis.

Insights from Community Detection Algorithms

This approach allows researchers to uncover **emerging trends and research hotspots** by examining how frequently keywords co-occur in academic literature.

2.9. FUTURE DIRECTIONS AND AREAS FOR IMPROVEMENT

Identifying Limitations

The paper [3] offers broad practical insights into sentiment analysis techniques while also pointing out limitations in current methods, such as low accuracy, data sparsity, and inability to capture subtle sentiments.

Recommendations for Future Research

Suggested areas of improvement include:

- Handling **shifting sentiments** within sentences.
- Increasing **classification accuracy** by combining rule-based and learning-based methods.
- Integrating diverse data types (e.g., **text, images, audio**) for richer sentiment understanding.
- Addressing **cultural and language variations** to build more globally adaptable sentiment models.

2.10. CONCLUSION

Sentiment analysis is used to extract from text people's opinions and feelings about products, services, concepts, ideas, and so on, which could come in the form of posts in social media, reviews in blogs, or comments about some events. Techniques such as Naïve Bayes, SVM, and Maximum Entropy classify sentiment as well as possible, yet problems such as changes in the meaning of the sentiment, accuracy problems, and a lack of data persist. The sheer quantity of user-generated content online is an opportunity but also a challenge. New methodologies, such as keyword co-occurrence analysis and community detection, may hold promise, but much remains to be done in fully understanding sentiment in text.

3. RESULTS AND DISCUSSIONS

The sentiment analysis project utilized various machine learning models to classify customer reviews into three classes: positive, neutral, and negative sentiments. Key steps included preprocessing the text, balancing the dataset using SMOTE, and evaluating the performance of models based on accuracy, precision, recall, and

F1-score. The following sections discuss the findings and comparative performance of the models:

3.1. DATASET OVERVIEW AND PREPROCESSING

The dataset initially had missing values, which were handled by dropping incomplete rows. Reviews were cleaned by removing HTML tags, punctuation, and stop words, followed by lemmatization. Due to class imbalance (more positive reviews), balancing was performed using downsampling for the majority class and upsampling with SMOTE for the minority class.

TF-IDF vectorization was applied to convert text into numerical features, selecting the top 7,500 features based on frequency and importance.

3.2. MODEL PERFORMANCE

Several machine learning models were evaluated for sentiment classification:

3.2.1 Logistic Regression

- Accuracy: 79%
- Precision: 80%
- Recall: 79%
- F1-Score: 79%
- Training Time: 2 minutes

Logistic Regression emerged as the best-performing model, offering the highest accuracy and F1-score. Its training time was also significantly lower than that of SVM.

3.2.2 Support Vector Machine (SVM)

- Accuracy: 79%
- Precision: 80%
- Recall: 79%
- F1-Score: 79%
- Training Time: 353 minutes

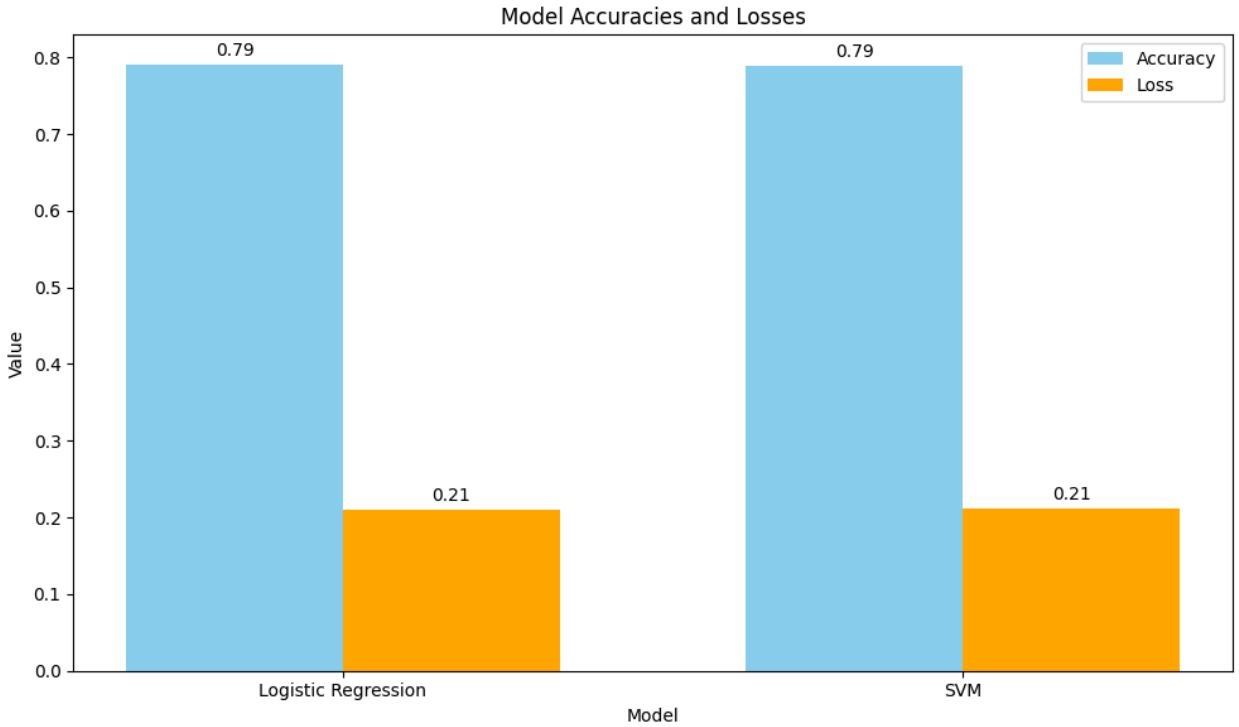
- Comparable accuracy to Logistic Regression but significantly higher training time.

3.2.3 KEY OBSERVATIONS

- Logistic Regression offered the best balance between performance and efficiency, ideal for real-time applications.
- SMOTE effectively improved model sensitivity to minority classes.
- Word Cloud visualization helped interpret important sentiment-contributing terms.

3.2.4 COMPARATIVE VISUALIZATION

- A bar chart compared model accuracies.
- A summary table presented accuracy, precision, recall, F1-score, and training times, highlighting trade-offs between performance and computational cost.



	Model	Accuracy	Precision	Recall	F1-Score	TrainingTime (mins)
0	Logistic Regression	79%	80%	79%	79%	2
1	SVM	79%	80%	79%	79%	353

4. CONCLUSION AND FUTURE WORK

4.1. CONCLUSION

This sentiment analysis project evaluated multiple machine learning models to classify customer reviews into positive, neutral, and negative sentiments. Among the models tested, **Logistic Regression** and **Support Vector Machine (SVM)** delivered the highest performance across all key metrics (accuracy, precision, recall, and F1-score), demonstrating strong classification capabilities.

However, **Logistic Regression** required significantly less training time **(2 minutes)** compared to **SVM (353 minutes)**, making it more efficient for real-time applications. Considering the trade-off between performance and computational cost, Logistic Regression was selected as the final model due to its scalability and practical deployment benefits.

4.2. FUTURE WORK

To further enhance sentiment analysis systems, future research can focus on the following areas:

- **Context-Aware Sentiment Detection:** Improve handling of polarity shifts and context-dependent sentiment, where the same words may imply different sentiments based on context.
- **Handling Ambiguity and Data Sparsity:** Develop better strategies to manage ambiguous expressions and sparse data, particularly in short or informal reviews.

- **Deep Learning Approaches:** Explore neural network-based models such as LSTM, BERT, or Transformers, which may capture deeper linguistic nuances and improve classification accuracy.
- **Multimodal Sentiment Analysis:** Integrate additional data sources (images, audio, video) alongside text, especially for applications like social media sentiment tracking.
- **Keyword Co-occurrence and Trend Detection:** Use co-occurrence networks and community detection algorithms to uncover emerging sentiment trends and hidden patterns.
- **Cross-Cultural and Multilingual Insights:** Incorporate language-specific and cultural context to improve global applicability and fairness of sentiment classification.

By addressing these challenges, future systems can become more accurate, context-aware, and scalable, enabling advanced sentiment analysis across a wide range of domains and platforms.

4.3. References

[1] A. M. Abirami and V. Gayathri, "A survey on sentiment analysis methods and approach," 2016 Eighth International Conference on Advanced Computing (ICoAC), Chennai, India, 2017, pp. 72-76, doi: 10.1109/ICoAC.2017.7951748. keywords: {Sentiment analysis;Ontologies;Feature extraction; Dictionaries; Blogs; Data mining; Data analysis; Data Analytics; sentiment Analysis; Decision making}

[2] N. Sultana, P. Kumar, M. R. Patra, S. Chandra, and S. K. S. Alam, “Sentiment Analysis for Product Review,” ICTACT Journal on Soft Computing, vol. 9, no. 3, pp. 1913-1919, Apr. 2019, doi: 10.21917/ijsc.2019.0266.

[3] Cui, J., Wang, Z., Ho, SB. et al. “Survey on sentiment analysis: evolution of research methods and topics.” Artif Intell Rev56, 8469–8510 (2023). https://doi.org/10.1007/s10462-022-10386-z