ALBUKHARY INTERNATIONAL UNIVERSITY

Group: 18
Lecturer Name: Prof. Ts. Dr. Nik Adilah Hanin Zahri
Course Name: Natural Language Processing
Course Code: CCS3153
Programme: Bachelor of Computer Science (Honours)
Year: 3
Semester: 1
Academic Year: 2025/2026
Project Title: Amazon Product's Positive and Negative Review classification

Name	Student Id	Group
MD ALTAF HOSSAIN SUNNY	AIU 222102182	D
Abdullahi Yasin Hersi	AIU21102400	C
Mohamed Abdullahi Ali	AIU22102342	D

Table of Contents

Introduction:

1.1 Introduction to the corpus:

The Amazon Product Reviews dataset is used for the purpose of sentiment analysis in this research. The dataset comprises 19,996 customer reviews of different products that were obtained from the Kaggle dataset in 2024. This ensures that the dataset used in the research meets the requirement of using the corpus within the previous five years. The dataset is composed of customer reviews in raw form written by the customer with the respective customer review having an attached customer review sentiment. Sentiment values are indicated in binary polarity, in which positive comments are labeled with 1 and negative comments with 0. Observations made from the dataset indicate an imbalance in the classes with respect to the sentiment of the comments, with 15,230 comments (76.17%) labeled as positive and the remaining 4,766 (23.83%) labeled as negative, since in actuality, consumers tend to show more affinity to positive comments than negative ones when carrying out electronic transactions.

The corpus encompasses various consumer opinions, experiences, and emotional expressions about different Amazon products. As such, the corpus encompasses quite rich linguistic features that are significant for sentiment classification tasks. The differing lengths of the reviews, vocabularies, as well as the levels of intensity in the opinions, further make the corpus suitable for research in Natural Language Processing. With regard to its size, authenticity, and the presence of emotional and rich-content material, the Amazon Product Reviews Corpus can be considered an optimal foundation for model building and comparisons of machine learning models. It also plays an important part in analyzing customer sentiment and building sentiment classification systems with positive and negative opinions.

1.2 Objective

The major aim of the project is to design, implement, and assess an NLP application to determine the polarity of the sentiments of consumer product reviews. This system should automatically classify customers' opinions as positive or negative according to textual review data.

The objectives of this project are specifically to:

- Apply appropriate NLP preprocessing to clean, process, and represent the textual review data.
- Implement and compare two machine learning algorithms for supervised learning: Logistic Regression and Random Forest, for the task of sentiment classification.
- Evaluate the performance of the models using appropriate metrics.
- Identify the best model for the most accurate detection of customer sentiment.
- Demonstrate how sentiment analysis can support business decision-making by providing insights into customer opinions and reactions.

1.3 Scope of Task

The scope of this sentiment analysis task is defined as follows:

- **Domain**: Consumer product reviews from the Amazon platform.
- **Task Type**: Binary sentiment classification (Polarity detection: Positive review=1, and Negative Review = 0).
- **Data Limit**: Analysis is performed on a subset of nearly 20,000 recent reviews to ensure computational efficiency and relevance.
- **Techniques**: The study is limited to traditional machine learning classifiers (Logistic Regression and Random Forest) using TF-IDF vectorization, excluding deep learning architectures or real-time API deployments.

2 Frame work design and model justification

2.1 Data preprocessing

Before the training of machine learning models, an extensive preprocessing of raw reviews from the Amazon products was performed in an ordered and methodological manner. The key aim of such preprocessing was noise reduction and improvement in the quality of features created from the raw reviews in the sentiment classifications. The procedures involved in preprocessing can be briefly outlined below:

Lower casing: all review texts were transformed to lowercase so as to make sure that there were no variations in tokenization. This allows the model not to treat features like "Excellent" and "excellent" as distinct features, thus eliminating redundancy in features.

Tokenization: the pre-cleaned reviews were tokenized for individual words using the word_tokenize function from the NLTK library. The mechanism of tokenizing helps in processing words in text in the model, which is significant for feature extractions using TF-IDF.

Stopword Removal: in the tokenization process, common stop words in the English language like "the," "is," and "and" were removed using the stop word list provided in the NLTK library. These words often occur in documents but do not provide significant information on polarity.

Lemmatization: was conducted by utilizing the WordNetLemmatizer in removing the last vowel of words and reducing them into base or root words. Words such as 'reviews' would be lemmatized into 'review'. The steps improve the merging of semantically equivalent words and also decrease model dimensionality.

2.2 Feature engineering

TF-IDF The **TF-IDF (Term Frequency-Inverse Document Frequency)** vectorizer was chosen to transform text into numerical features. This method was selected because:

- It penalizes words that appear too frequently across the entire corpus (like "product"), giving higher weight to words that are rare but sentiment-rich.
- It creates a high-dimensional sparse matrix that captures the unique linguistic fingerprint of each review.

2.3 Choice of Machine Learning Algorithms

Two supervised machine learning algorithms, Logistics Regression and Random Forest, have been chosen for this analysis because of their proven ability in performing text classification tasks along with their relative strength in dealing with the problem of sentiment analysis.

Justification for Logistic Regression:

Baseline Performance: The baseline performance uses a logistic regression classifier as a baseline, which efficiently performs a linear classification for binary problems, such as text categorization,

Interpretability: It generally has an easier explanation between word weights and the sentiment because one can clearly determine which word ("disappointed" or "amazing") the classification distinction depends on.

Efficiency: It is highly efficient when handling high-dimensional sparse data created by TF-IDF without the need for much computational power.

Justification for Random Forest:

Dealing with Non-linearity: Random Forest is an ensemble of Decision Trees that can handle the non-linear interaction between the words in the review, which is not possible with Logistic Regression.

Robustness to Overfitting: Since Random Forest involves several decision trees and feature selections, it keeps its model less prone to overfitting to particular “noisy” comments, unlike decision tree techniques.

Feature Importance: This makes it possible to identify which words were most important to come to a decision in the forest decision-making processes.

2.4 Evaluation Method

To ensure a fair comparison, the models are evaluated using Accuracy and the F1-Score. The F1-Score is particularly critical here because the dataset is imbalanced (76% positive), and F1 provides a better measure of how well the model handles the minority (negative) class.

3. Implementation/Coding

3.1 Technical Environment

The system was developed using **Python 3.10** in a Jupyter Notebook environment. Key libraries utilized include:

- **Pandas**: For data manipulation and CSV handling.
- **NLTK**: For specialized Natural Language Processing tasks like tokenization and lemmatization.
- **Scikit-Learn**: For implementing TF-IDF vectorization and the machine learning classifiers.

3.2 Core Functionality and Code Logic

Here implementation follows a modular structure to ensure the system works as intended without errors:

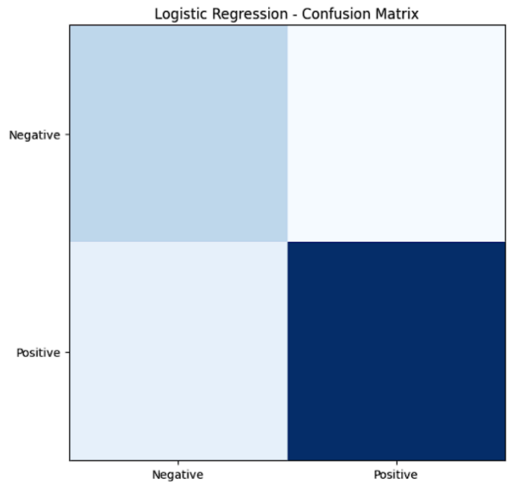
- **Data Loading**: The amazon.csv file is loaded into a Pandas DataFrame, and initial checks for null values are performed to ensure data integrity.
- **Preprocessing Pipeline**: A dedicated function or series of cell operations handles text cleaning. For instance, the code uses WordNetLemmatizer to reduce lexical variety and focus on core sentiment tokens.
- **Vectorization Strategy**: The TfidfVectorizer is configured to transform the processed reviews into a numerical feature matrix, which is then split into training (80%) and testing (20%) sets to validate the models.

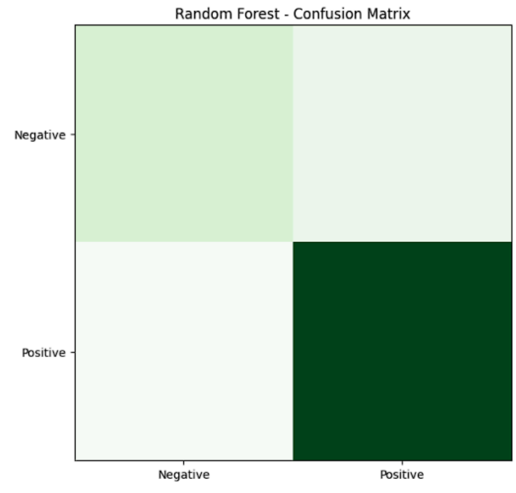
3.3 Machine Learning Execution

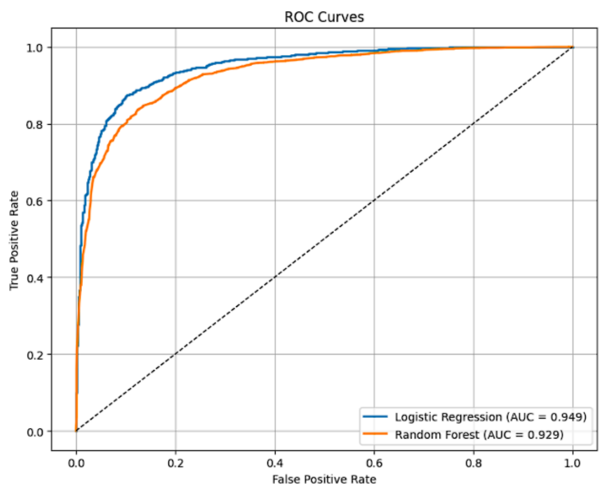
The system executes two distinct modeling paths:

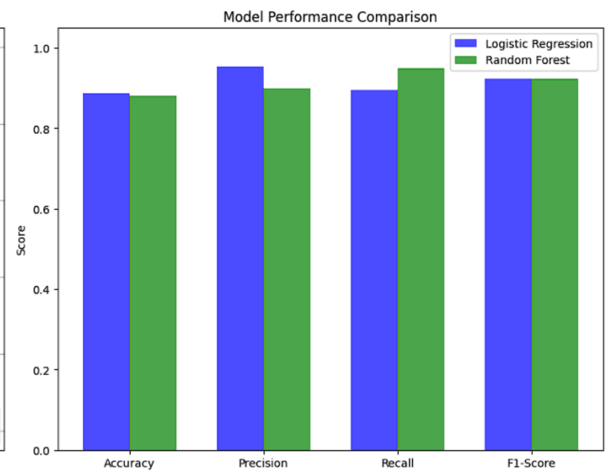
- **Linear Path (Logistic Regression)**: A LogisticRegression object is instantiated and fitted to the training data. This serves as our high-speed, interpretable baseline.
- **Ensemble Path (Random Forest)**: A RandomForestClassifier with multiple estimators is trained. This allows the system to capture complex patterns within the reviews that a linear model might miss.

3.4 Graphical Representation of Results









MODEL PERFORMANCE COMPARISON:

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	0.88675	0.954132	0.894322	0.923259	0.949286
Random Forest	0.88100	0.899844	0.949458	0.923986	0.929261

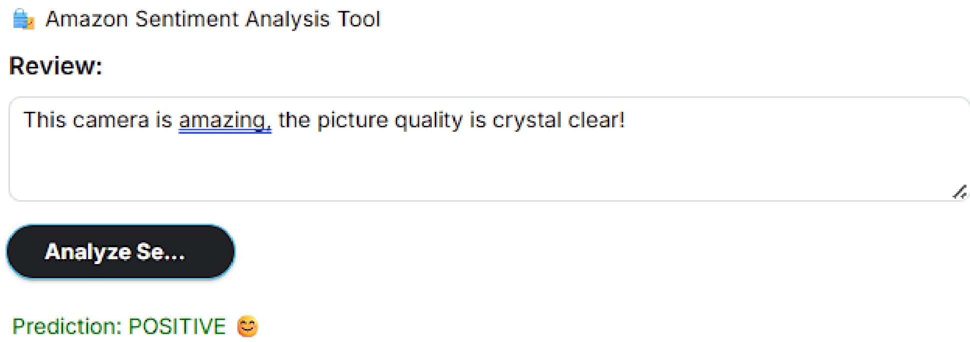
4. Demonstration of the NLP Application:

Application Overview: The application is a Python-based interactive tool developed using ipywidgets. It serves as a real-time sentiment classifier that allows users to input raw Amazon product reviews and receive an instant polarity prediction (Positive or Negative).

Technical Integration: The software backend is powered by a **Random Forest** model and a **TF-IDF Vectorizer** (5,000 features, including bigrams) that were serialized using joblib.

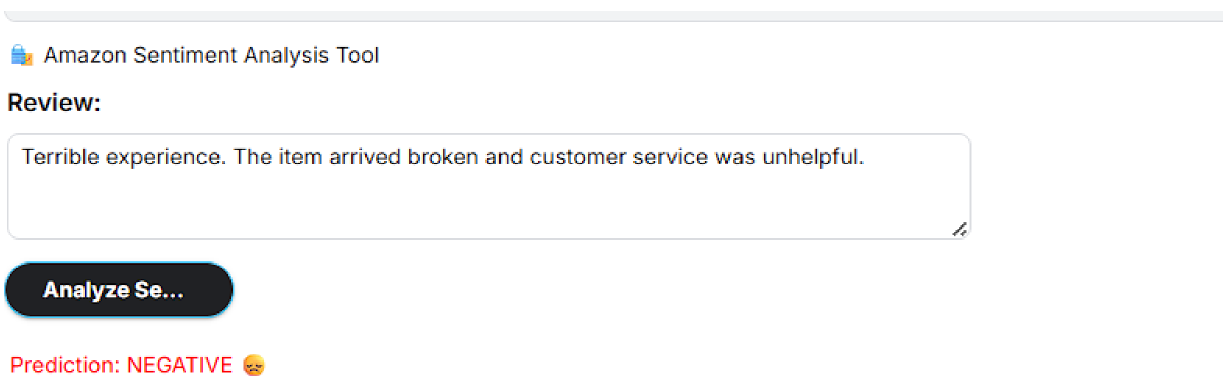
Test Case 1 (Positive):

- *Input:* "This camera is amazing, the picture quality is crystal clear!"
- *Output:* **POSITIVE** 😊



Test Case 2 (Negative):

- *Input:* "Terrible experience. The item arrived broken and customer service was unhelpful."
- *Output:* **NEGATIVE** 😞



Robustness: The application demonstrates robustness by handling special characters and casing through a specialized preprocessing pipeline integrated directly into the widget logic.

5 Conclusion:

This project was able to design and test the performance of the system for the sentimental analysis of around 20,000 product reviews on the Amazon website with a properly designed Natural Language Processing pipeline. Normal processing methods, TF-IDF method processing, and machine learning models have been used to classify the opinions of consumers into negative and positive sentiments. Two machine learning models, namely Logistic Regression and Random Forest, were coded and compared on the basis of various parameters such as accuracy, precision, recall, F1 score, and ROC AUC value. Both models performed very well on each parameter, but the Logistic Regression model showed better accuracy, AUC value, precision on imbalanced datasets, and cross-validation scores. In addition to these, the model was more efficient and easier to interpret.

Error analysis showed that misclassifications were due to mixed or neutral sentiment expressions in the reviews, revealing a common limitation of the TF-IDF-based models. Given this limitation, the system performs reliably on clearly positive and negative reviews. The final product of the project will be a functional and robust sentiment analysis application, including an interactive ipywidgets-based interface capable of making real-time sentiment predictions. Overall, this project meets the course learning outcomes CLO2 and CLO3 and presents an efficient, scalable methodology that can be adopted for the automation of consumer sentiment analysis in support of business decision-making.