

**DEPARTMENT OF ROBOTICS & ARTIFICIAL INTELLIGENCE**

**Total Marks:** 7.5

**Obtained Marks:** \_\_\_\_\_

# **LAB: Programming For Artificial Intelligence**

## **Sentiment Analysis on Amazon Customer Reviews Using Machine Learning**

### **Group Members:**

<b>Submitted To:</b>	<b>Sir Anees Tariq</b>
<b>Enrollment</b>	<b>Name</b>
23108328	Talha Ahmed Khan
23108291	Arzoo Sarwari
23108281	Abdul Rehman

## 1. Abstract

This project aims to perform sentiment analysis on Amazon product reviews to determine whether a customer's feedback is **positive, negative, or neutral**. The dataset contains thousands of customer reviews, including textual feedback and numerical ratings. Using **Natural Language Processing (NLP)** and **machine learning algorithms**, this project preprocesses textual data, extracts features using TF-IDF, and evaluates multiple models to identify the best-performing classifier. A web-based application will also be developed using **Streamlit** to provide an interactive interface where users can input reviews and instantly receive sentiment predictions.

## 2. Background Problem

With the exponential growth of e-commerce platforms, customers frequently rely on reviews to make purchasing decisions. However, analyzing thousands of reviews manually is time-consuming and prone to human bias. Retailers also struggle to extract meaningful insights from such large amounts of textual feedback. Therefore, automating the sentiment classification of customer reviews can help businesses improve product quality, monitor customer satisfaction, and make data-driven decisions efficiently.

## 3. Introduction

Sentiment Analysis, also known as **opinion mining**, is a key task in **Natural Language Processing (NLP)** that involves determining the emotional tone behind textual data. This project leverages NLP and machine learning techniques to classify Amazon product reviews as **positive, negative, or neutral**.

The model is trained using real-world Amazon review data and later integrated into a **Streamlit-based web application** for easy access and deployment. This project demonstrates the complete workflow of an AI-based text classification pipeline — from preprocessing to model deployment.

## 4. Objectives

- To preprocess and clean Amazon product reviews using NLP techniques.
- To perform **exploratory data analysis (EDA)** for understanding data distribution.
- To train and evaluate multiple machine learning models for sentiment classification.
- To identify the most accurate model based on performance metrics such as accuracy, precision, recall, and F1-score.
- To build and deploy an interactive **Streamlit web app** for real-time sentiment prediction.

## 5. Methodology

The project follows a **systematic pipeline** consisting of the following steps:

### 1. Data Collection

The dataset used is *Reviews.csv* containing Amazon product reviews with features such as *ProductId*, *Score*, *Summary*, and *Text*.

## 2. Data Preprocessing

- Handling missing values
- Cleaning text (removing punctuation, HTML tags, URLs, and stopwords)
- Converting text to lowercase

```
def clean_text(text):  
    text = text.lower()  
    text = re.sub(r"http\S+", "", text)  
    text = re.sub(r"<.*?>", "", text)  
    text = re.sub(r"^[^a-z\s]", "", text)  
    text = re.sub(r"\s+", " ", text).strip()  
    return text
```

## 3. Feature Extraction

Text is transformed into numerical form using **TF-IDF Vectorization** to represent important words.

```
vectorizer = TfidfVectorizer(max_features=5000,  
                            ngram_range=(1, 2))  
X_train_tfidf = vectorizer.fit_transform(X_train)  
X_test_tfidf = vectorizer.transform(X_test)
```

## 4. Model Training and Evaluation

Various machine learning models were tested:

- Logistic Regression
- K-Nearest Neighbors
- Decision Tree
- Naive Bayes
- Random Forest

## 5. Model Selection & Deployment

The best-performing model (based on accuracy) will be saved using **Pickle** and integrated into a **Streamlit application** for deployment.

## 6. Tools and Technologies

- **Programming Language:** Python
- **Libraries & Frameworks:**
  - pandas, numpy, matplotlib, seaborn
  - scikit-learn (for model training and evaluation)
  - re, string (for text cleaning)
  - WordCloud (for visualization)
  - Streamlit (for web app deployment)
- **Development Environment:** Jupyter Notebook, Streamlit Cloud
- **Dataset:** Amazon Product Reviews Dataset (Reviews.csv)

## 7. Components Used

- **Data Processing Component:** Handles text cleaning, tokenization, and formatting.
- **Feature Extraction Component:** Uses TF-IDF to convert text into numeric vectors.
- **Model Training Component:** Applies various ML algorithms to classify sentiment.
- **Evaluation Component:** Compares model accuracy and visualizes results.
- **Deployment Component:** Streamlit app for user interaction and prediction.

## 8. Code

```
import pandas as pd
import numpy as np
import re

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report

df = pd.read_csv("Reviews.csv")
df.dropna(subset=['Text', 'Score'], inplace=True)

def get_sentiment(score):
    if score <= 2:
        return 'negative'
    elif score == 3:
        return 'neutral'
    else:
        return 'positive'

df['Sentiment'] = df['Score'].apply(get_sentiment)
X_train, X_test, y_train, y_test = train_test_split(df['Text'],
df['Sentiment'], test_size=0.2)

vectorizer = TfidfVectorizer(max_features=5000)
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)

model = LogisticRegression(max_iter=200)
model.fit(X_train_tfidf, y_train)
y_pred = model.predict(X_test_tfidf)
```

```
print("Accuracy:", round(accuracy_score(y_test, y_pred)*100, 2),  
      "%")
```

## 9. Expected Outcomes

- A well-trained sentiment analysis model capable of classifying reviews with **high accuracy**.
- Visual insights on review distribution and word frequency using WordClouds and graphs.
- A **Streamlit-based web application** allowing users to input new reviews and instantly view sentiment predictions.
- Exported trained model and vectorizer for future reuse.

## 10. Conclusion

This project demonstrates how **machine learning** and **NLP** can be effectively combined to extract valuable insights from textual data. It automates the sentiment classification process for e-commerce reviews, saving time and enhancing decision-making for businesses. Among the tested models, **Logistic Regression** and **Naive Bayes** are expected to yield the best balance between accuracy and computation time.

## 11. Future Enhancement

- Implement **Deep Learning models** such as **LSTM** or **BERT** for improved accuracy.
- Add **multilingual support** to handle reviews in various languages.
- Integrate **real-time review scraping** from Amazon APIs.
- Improve visualization with **interactive dashboards** in Streamlit.
- Deploy the app using **Docker** or **AWS EC2** for better scalability and performance.

## 12. References

- Kaggle Amazon Review Dataset: <https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews>
- Scikit-learn Documentation: <https://scikit-learn.org/>
- Streamlit Documentation: <https://docs.streamlit.io/>