BAHRAIN POLYTECHNIC بوليتكنك البحرين

Assessment Cover Sheet

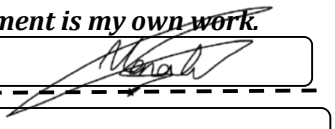
Assessment	Project(Individual)		
Assessment	Uncontrolled	Individual	Not must-pass
Due Date	11 December 2025	**Course Code**	IT9002
Course Title	Natural Language Processing		
Internal Moderator's	Dr. Abdelhameed Fawzy		
External Examiner's	Dr.		

Instructions:

1. This cover sheet must be completed (section in blue below) and attached to your assessment before submission in hard copy/soft copy.
2. The time allowed for this assessment is XXX minutes/hours/days.
3. This assessment carries XXX marks distributed to a total of XXX questions assessing CILO X and CILO X.
4. The materials allowed for use in this assessment are XXX, XXX, and XXX.
5. The **use of generative AI tools is strictly prohibited**.
6. References consulted (if any) must be properly acknowledged and cited.
7. The assessment has a total of XXX pages

Learner ID	12011218	**Date Submitted**	8th January 2026
Learner Name	Manal Talal Ameen		
Programme	MSc in AI		
Programme	IT9002		
Lecturer's	Sini Raj Pulari		

By submitting this assessment for marking, I affirm that this assessment is my own work.

Learner 

Do not write beyond this line. For assessor use

Assessor's Name	Sini Raj Pulari		
Marking Date		**Maks Obtained**	

Comments:

2026

Sentiment Analysis of Amazon Electronics Reviews Using Natural Language Processing

IT9002 NATURAL LANGUAGE PROCESSING

MANAL TALAL

BAHRAIN POLYTECHNIC | Masters in Ai

Table of Contents

List of Figures

Sentiment Analysis of Amazon Electronics Reviews Using Natural Language Processing

1. Introduction

Customer feedback on the internet is important in affecting consumer behavior on online shopping sites like Amazon. Automated processing of sentiment in substantial amounts of textual reviews facilitates business in knowing customer opinion, discovering strengths and weaknesses of products and aiding in decision-making that is based on facts. It is through the NLP techniques that it becomes possible to process text data that is unstructured effectively. This report presents an end-to-end NLP pipeline on sentiment analysis based on the Amazon Electronics Reviews dataset. The study applies these techniques to the Amazon data set: data loading, preprocessing, feature extraction, sentiment labelling, model training, evaluation, and discussion of results based on the classical machine learning methods.

2. Dataset Description

This research uses the Amazon Electronics Reviews data set, which is uploaded by Shivam Parab found on Kaggle. It consists of reviews of customers related to electronic items offered on Amazon.

Key attributes utilized:

- Review Text: The textual information of the customer review.
- Overall: Star rating provided by the customer (1 to 5).

The data is provided in the form of a JSON line (Electronics_5.json). Due to the large size dataset, only a subset of the data was loaded using a row limited during experimentation to model practically in Google Colab.

3. Data Loading and Exploration

The dataset was then loaded into Google Colab and by connecting it to Google Drive and processed using the Pandas library. The first stage in the exploratory study was to manually look at a sufficient amount of customer reviews to verify the data was loaded correctly and to explain the dataset's structure.

```
[2]
✓ 12s
import pandas as pd

path = "/content/drive/MyDrive/NLP/Report/Electronics_5.json"
df = pd.read_json(path, lines=True, nrows=200000)

df = df[['reviewText', 'overall']].dropna()
df.head()
```

	reviewText	overall
0	We got this GPS for my husband who is an (OTR)...	5
1	I'm a professional OTR truck driver, and I bou...	1
2	Well, what can I say. I've had this unit in m...	3
3	Not going to write a long review, even thought...	2
4	I've had mine for a year and here's what we go...	1

Figure 1: Sample of the Amazon Electronics Reviews dataset after loading.

Next, an exploratory study was done to look at how the star ratings were distributed in the sample records that had been looked at before.

The outcome of this study yields insights into the class imbalance and the prevailing sentiment shown in customer feedback.

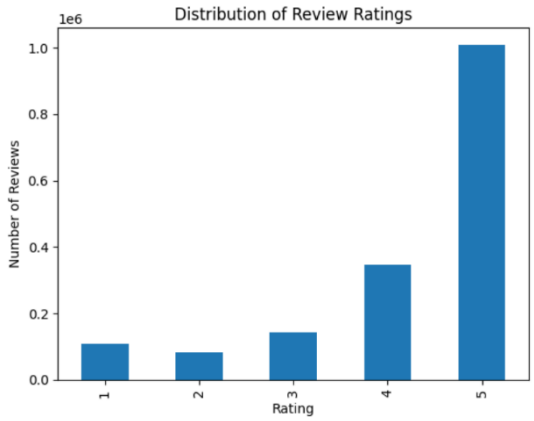


Figure 2: Distribution of star ratings in the Amazon Electronics Reviews dataset.

4. Text Preprocessing

Textual reviews frequently contain unnecessary components such as URLs, punctuation marks, and irregular upper case and lower-case letters. Series of preprocessing were implemented to improve model performance:
• Conversion to lowercase
• Removal of URLs
• Removal of non-alphabetic characters
• Removal of extra whitespace

```
import re

def clean_text(t):
    t = t.lower()
    t = re.sub(r"http\S+|www\S+", "", t)
    t = re.sub(r"[^a-z\s]", " ", t)
    t = re.sub(r"\s+", " ", t).strip()
    return t

df['clean'] = df['reviewText'].apply(clean_text)
df[['reviewText','clean']].head(2)
```

	reviewText	clean
0	We got this GPS for my husband who is an (OTR)...	we got this gps for my husband who is an otr o...
1	I'm a professional OTR truck driver, and I bou...	i m a professional otr truck driver and i boug...

Figure 3: Comparison of original review text and cleaned text.

5. Tokenization, Stemming, and Lemmatization

After the text cleaning stage in the preprocessing phase, further Natural Language Processing methods were used to transform the textual information into a more structured and consistent form that could be easily used in feature extraction. These methods included tokenization, stemming and lemmatization.

5.1 Tokenization

The process of tokenization was conducted to cut the cleaned review text into small separate units (tokens) using whitespace-based splitting. This step converts each review into a list of tokens.

```
df['tokens'] = df['clean'].str.split()
df['tokens'].head(2)
```

5.2 Stemming

Stemming was implemented using the porter stemmer, which returns words to their root form by removing common suffixes. This process has the effect of reducing the size of vocabulary though it can produce non-lexicalized tokens.

```
import nltk
from nltk.stem import PorterStemmer
stemmer = PorterStemmer()
df['stemmed'] = df['tokens'].apply(lambda toks: [stemmer.stem(w) for w in toks])
df[['tokens','stemmed']].head(2)
```

5.3 Lemmatization

WordNet Lemmatizer was used to Lemmatize words into recognized dictionary forms. Lemmatization, in contrast to stemming, produces linguistically sound words and does not take away the sense.

```
from nltk.stem import WordNetLemmatizer
lemm = WordNetLemmatizer()
df['lemmatized'] = df['tokens'].apply(lambda toks: [lemm.lemmatize(w) for w in toks])
df[['tokens','lemmatized']].head(2)
```

```
df[['tokens', 'stemmed', 'lemmatized']].head(2)
```

	tokens	stemmed	lemmatized
0	[really, i, see, people, complaining, more, ab...	[realli, i, see, peopl, complain, more, about,...	[really, i, see, people, complaining, more, ab...
1	[i, had, to, return, the, first, one, due, to,...	[i, had, to, return, the, first, one, due, to,...	[i, had, to, return, the, first, one, due, to,...

Figure 4: Comparison of tokenization, stemming, and lemmatization results for sample reviews.

6. Feature Extraction

After text processing and normalization, the numeric features were obtained using textual data to enable machine-learning models to be trained. Since machine-learning algorithms do not take raw textual data as input, the cleaned reviews have been transformed into numeric data through the Term Frequency-Inverse Document Frequency (TFIDF) method.

TF-IDF places more emphasis on the words that are highly frequent in a particular review and are uncommon to the entire corpus, therefore allowing the model to focus on the most informative and discriminative lexical terms.

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf = TfidfVectorizer(max_features=5000)
X_tfidf = tfidf.fit_transform(df['final_text'])
X_tfidf.shape
```

7. Sentiment Label Creation

To ease the process of classifying sentiments under supervision, sentiment labels were derived based on the numeric ratings of stars they had within the dataset. The transformation to sentiment categories used the traditional thresholds that are commonly used in the industry of sentiment analysis.

This labeling strategy allowed the model to link consumer feedback with sentiment categories enabling it to formulate with a supervised learning task:
• Ratings 1-2 : Negative (0)
• Rating 3 : Neutral (1)
• Ratings 4-5 : Positive (2)

7.1 Sentiment Mapping Code

In this regard, the subsequent role was utilized to convert quantitative ratings of stars into categorical sentiment:

```
def label_sentiment(r):
    if r <= 2:
        return 0
    elif r == 3:
        return 1
    else:
        return 2

df['sentiment'] = df['overall'].apply(label_sentiment)
df['sentiment'].value_counts()
```

sentiment	count
2	40287
0	5728
1	3985

dtype: int64

7.2 Visualization of Sentiment Class Distribution

The proportion of the number of sentiment classes present in the dataset was explored by visualizing the output of the labeled sentiment. The analysis sheds light on the possibility of class imbalance, which is a critical factor when supervising machine learning models, because imbalanced classes can lead to biased model behavior to the majority class.

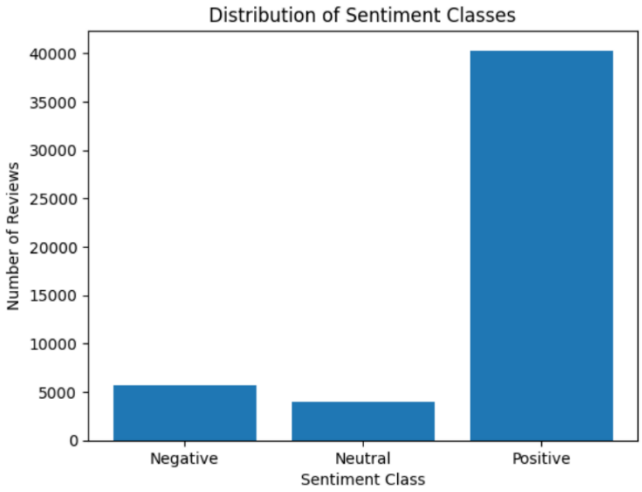


Figure 5: Categorical distribution of sentiment based on star rating data through inference.

8. Train-Test Split

After the extraction of the features, the study divided the dataset into a training and a testing set using an 80 20 split. The explanatory variables were the TF -IDF feature matrix, and the dependent variable was the sentiment annotations.

Logistic regression model was driven by the fact that it is a computationally efficient and effective algorithm in cases of text classification with a high-dimensional sparse representation. The model was trained using the training partition and then evaluated using the held back testing data.

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

X_train, X_test, y_train, y_test = train_test_split(
    X_tfidf, df['sentiment'], test_size=0.2, random_state=42
)

model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)
```

9. Model Training

The sentiment classification task was trained on two classical models of machine learning:

- Multinomial Naive Bayes
- Logistic Regression

These models were selected due to their effectiveness and application in numerous text classification problems. The feature representation obtained in the previous step was the TF-IDF and was inputted in both models.

The models were trained and tested using a held-out test set to achieve the performance of the models. Measures of standard classification such as accuracy, precision, recall, and F1 -score were used. Moreover, the distribution of correct and incorrect predictions was also analyzed using a confusion matrix to determine the distribution of the correct and incorrect predictions by the sentiment classes.

This comparison has made it possible to comparatively evaluate the performance of models and identify the effect of imbalance in classes on sentiment prediction.

```
from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score

y_pred = model.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
confusion_matrix(y_test, y_pred)
```

```
Accuracy: 0.8542
              precision    recall  f1-score   support

           0       0.73      0.49      0.58      1098
           1       0.37      0.07      0.12       782
           2       0.87      0.98      0.92      8120

    accuracy                           0.85     10000
   macro avg       0.66      0.51      0.54     10000
weighted avg       0.82      0.85      0.82     10000

array([[ 533,   32,  533],
       [  95,   55,  632],
       [ 105,   61, 7954]])
```

Figure 6: Confusion matrix of sentiment classifications result

10. Model Evaluation

The model performance was assessed using standard classification metrics including accuracy, precision, recall, F1-score . Multinomial Naïve Bayes and Logistic Regression were both explored; however, the overall performance of the latter was higher and therefor it was selected for detailed analysis.

The classification report for the Logistic Regression model highlights strong predictive performance for the positive sentiment class, while the neutral sentiment class exhibits low recall, likely due to class imbalance within the dataset.

The confusion matrix clarifies the number of correct and incorrect predictions there were by the model in each sentence category in relation to the sentiment; hence, providing information on the strengths and weaknesses of the model to predict one sentiment class versus another.

	precision	recall	f1-score	support
0	0.73	0.49	0.58	1098
1	0.37	0.07	0.12	782
2	0.87	0.98	0.92	8120
accuracy			0.85	10000
macro avg	0.66	0.51	0.54	10000
weighted avg	0.82	0.85	0.82	10000

Figure 7: Standard classification report of logistic regression

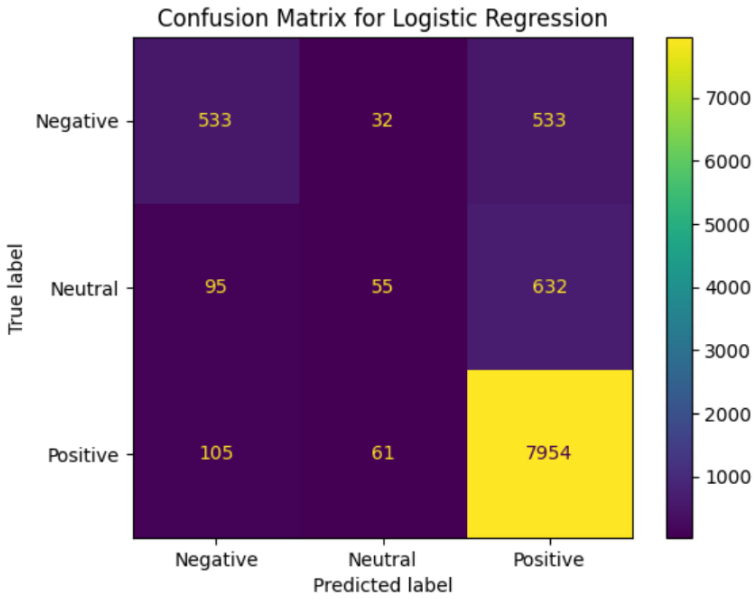


Figure 8: Confusion matrix for the logestic regression model

11. Bigram and Part-of-Speech (POS) Analysis

In order to further investigate the linguistic patterns of the dataset of Amazon Electronics Reviews, the frequency analysis of the bigrams and Part-of-Speech (POS) tagging were conducted.

11.1 Bigram Frequency Analysis

CountVectorizer was used to identify the most frequent two-word (bigram) sequences in the dataset with the help of the Bigram analysis. The findings indicate that common functional phrases like on the, in the, of the and it is predominate in the most common bigrams.

This should occur in large-scale natural language text since such combinations are representative of normal grammatical structure but not meaning that relates to the product. Nevertheless, frequency list may still offer a helpful overview of repetitive phrases patterns that are present in customer reviews.

1 to 20 of 20 entries Filter

index	Frequency
of the	20846
on the	14930
if you	14123
in the	13869
it is	12379
with the	11802
for the	11417
to the	11400
and the	11304
this is	10760
and it	8794
you can	8612
to be	7796
to use	6456
have to	6100
to get	5514
is the	5514
it wa	5462
the price	5336
but it	5249

Figure 9: Most frequent bigrams extracted from the review text using CountVectorizer

11.2 Part-of-Speech (POS) Tagging

POS tagging was used to analyze the grammatical structure of the customer reviews with the help of the NLTK tagging functionality. In POS tagging, a grammatical category is attached to every token (e.g., noun, verb, adjective). This can be used to find out how sentiment is represented linguistically, e.g. adjectives may be used to describe the quality of products (e.g. great, bad, excellent) and nouns may be used to describe the product features (e.g. battery, screen, cable). The tagged text offers a better insight into the value of various types of words to meaning and sentiment in review text.

```
[nltk_data]       date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[('really', 'RB'),
 ('i', 'JJ'),
 ('see', 'VBP'),
 ('people', 'NNS'),
 ('complaining', 'VBG'),
 ('more', 'RBR'),
 ('about', 'IN'),
 ('vonage', 'NN'),
 ('issue', 'NN'),
 ('than', 'IN'),
 ('this', 'DT'),
 ('router', 'JJ'),
 ('year', 'NN'),
 ('ago', 'IN'),
 ('configuration', 'NN'),
 ('wa', 'VBD'),
 ('a', 'DT'),
 ('nightmare', 'NN'),
 ('now', 'RB'),
 ('is', 'VBZ'),
 ('alot', 'RB'),
```

Figure 10: Example of Part-of-Speech (POS) tagging output for a sample review using NLTK

12. Discussion

The findings provide strong support that the classical machine learning models with TF-IDF features are useful in sentiment classification. Logistic Regression was also strongly performing especially on the positive sentiment which is not surprising as the dataset is highly dominated by positive reviews. Nevertheless, being neutral was difficult to categorize as precise, mostly because of the issue of class imbalance and a lack of linguistic differentiation between neutral and other sentiment categories. Such results reveal the significance of the data distribution, as well as feature representation in the application of the supervised sentiment analysis models.

13. Limitations

The research has several limitations. First, the dataset has an unbalanced classes distribution with a predominant number of positive reviews that affected the model performance. Second, the neutral sentiment was very hard to categorize because of the linguistic similarity with positive and negative reviews. Third, only classical models of machine learning were considered, and more sophisticated deep learning systems were not discussed. These weaknesses are indicative of future opportunities.

14. Conclusion

In this project, a complete NLP pipeline was used to analyze the sentiment of Amazon Electronics reviews. Customer feedback provided a valuable insight into the customer through preprocessing, feature extraction, and machine learning classification. The findings demonstrate the usefulness of classical NLP methods in real-world sentiment analysis. Future research may work on deep learning models and data balancing algorithms to enhance performance, especially in neutral sentiment classification.

References

Cambria, E., Poria, S., Bajpai, R., & Schuller, B. (2017). Sentiment analysis: Beyond positive and negative. *IEEE Intelligent Systems, 32*(4), 88–93.

Géron, A. (2022). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. *O'Reilly Media.*

Jurafsky, D., & Martin, J. H. (2023). *Speech and language processing.* Pearson.

Liu, B. (2020). Sentiment analysis: Mining opinions, sentiments, and emotions. *Cambridge University Press*.

Parab, S. (2020). *Kaggle*. (Amazon Electronics Reviews Dataset) Retrieved from https://www.kaggle.com/search

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.