Amazon Product Review Analysis

A CS2011 (Introduction to AI and ML) Project

Dr. Sobhan Kanti Dhara, Instructor

Capacity Building for Design and Entrepreneurship (CBDE) Program

ECE & EIE Department, National Institute of Technology, Rourkela

Submitted: December 3, 2024

Project Objective:

The objective of this project is to develop an ML-based program to analyze Amazon product reviews efficiently and automate the process to address several challenges faced by both customers and businesses. The ultimate goal is to create a tool that empowers customers and businesses alike by making review data accessible, trustworthy, and actionable.

Problem Analysis & Solution

Challenges Faced

- Overwhelming volume of reviews
- Mixed and ambiguous sentiments
- Presence of fake reviews that reduce trust
- Difficulty in extracting actionable insights for businesses

Solution

Develop an automated NLP and ML-based system to process, analyze, and visualize reviews effectively. This system will tackle the challenges by automating key aspects of review analysis, providing valuable insights for both customers and businesses.

Project Goals and Features

1

Aspect-Based Sentiment Analysis (ABSA)

Extract specific product aspects mentioned in reviews and determine associated sentiments (positive, negative, or neutral).

2

Review Summarization

Generate concise summaries of reviews by extracting essential points.

3

Spam Detection

Identify and filter fake, irrelevant, or malicious reviews using text patterns and metadata.

4

Trend Analysis

Analyze trends in customer satisfaction over time for specific products or categories.

5

Visualization and Reporting

Provide interactive dashboards with features like word clouds, sentiment graphs, and aspect-based insights.

Dataset and Preprocessing

Data Source

- Amazon product review datasets available online
- Kaggle Datasets: https://www.kaggle.com/datasets/arhamrumi/amazon-product-reviews/datahttps://www.kaggle.com/datasets/jillanisofttech/amazon-product-reviews
- Data Format: CSV files with columns like review text, ratings, product categories, and metadata

Preprocessing

- Handle missing values
- Remove stop words, punctuation, and special characters
- Normalize text (e.g., stemming, lemmatization)

Methodology: Data Preprocessing

Cleaning

Remove duplicates, irrelevant characters, and non-textual data.

Tokenization

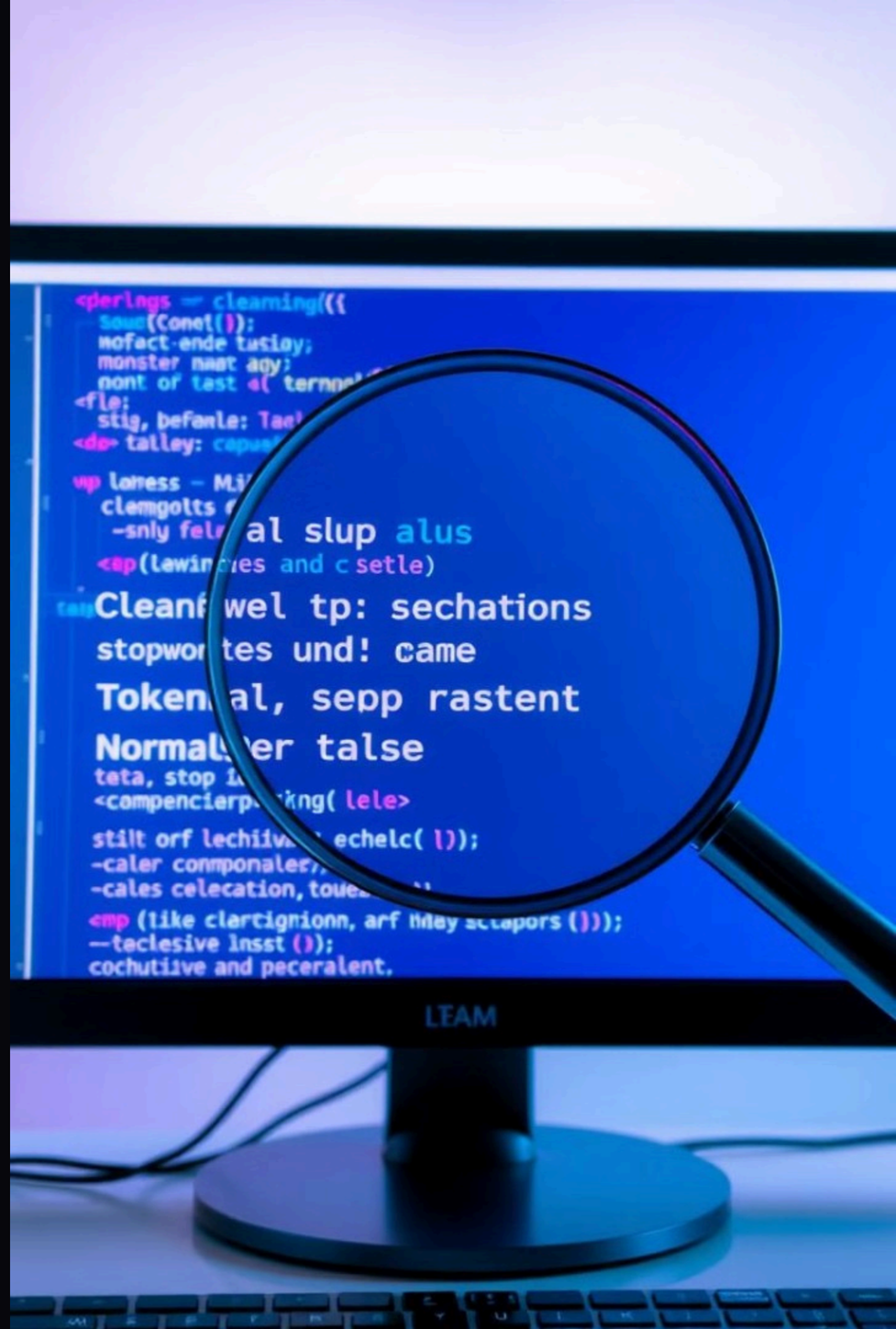
Break reviews into words or phrases for analysis.

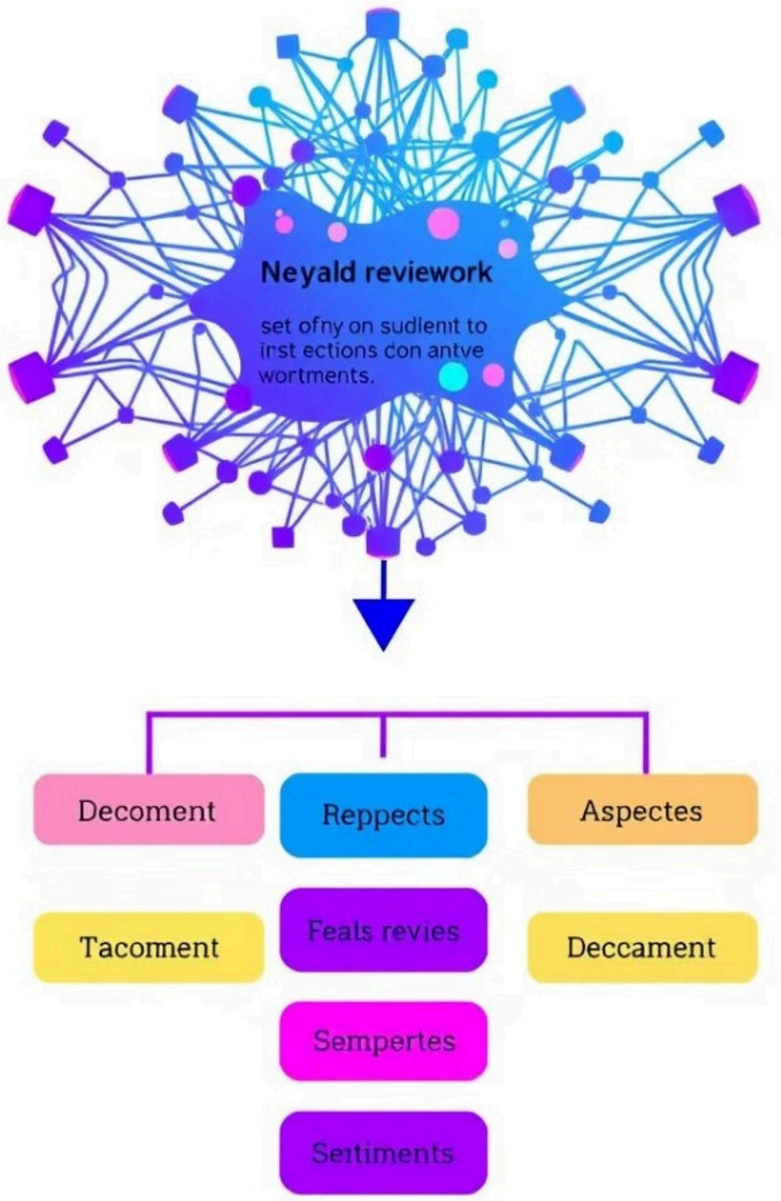
Stopword Removal

Filter out commonly used words (e.g., "and," "the").

Multilingual Handling

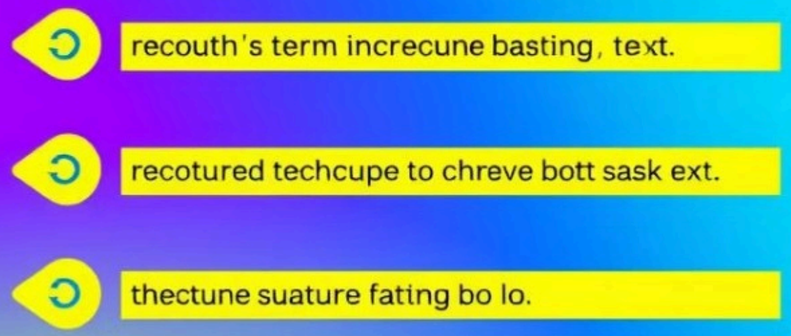
Use translation APIs or multilingual NLP models.





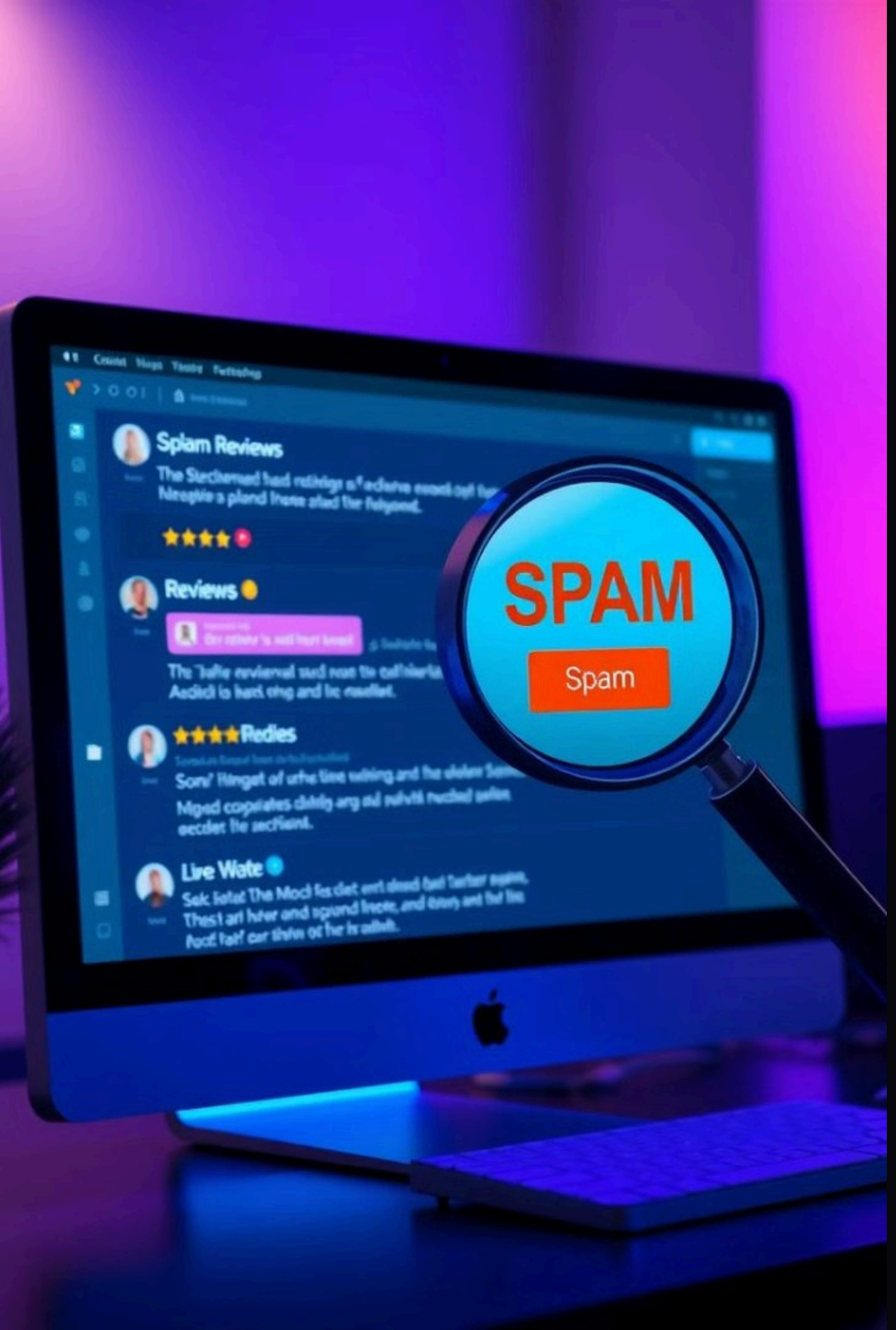
Methodology: Aspect-Based Sentiment Analysis

Utilize pretrained NLP models like BERT, RoBERTa, or OpenAI APIs, or fine-tune custom models for aspect extraction.



Methodology: Review Summarization

Build summarization models with libraries like BART, T5, or Sumy. These models can extract essential information from lengthy reviews, creating concise summaries.



Methodology: Spam Detection

Train classification models with labeled data to detect spam. Features to consider: review length, keyword frequency, and metadata.

Expected Impact

For Customers

- Simplified decision-making through categorized and summarized reviews.
- Trustworthy recommendations by filtering fake reviews.

For Businesses

- Better insights into product performance and customer feedback.
- Enhanced strategies based on detailed trend analyses.

Tools and Technologies

Programming Language: Python (for flexibility and ML/NLP libraries).

Libraries:

NLP: SpaCy, TextBlob, NLTK, Hugging Face Transformers.

Visualization: Matplotlib, Seaborn, Plotly, Dash. ML Frameworks: Scikit-learn, TensorFlow, PyTorch. Dashboards: Dash, Streamlit, or Tableau. Data Handling: Pandas, NumPy.

Challenges

a. Mixed Reactions in Reviews: Develop models that can identify nuanced sentiments.

b. Multilingual Reviews: Incorporate multilingual NLP models like mBERT.

c. Quality and Scale of Data: Ensure the dataset is clean and balanced to prevent model biases.Expected Impact

EXPECTED IMPACT

For Customers: Simplified decision-making through categorized and summarized reviews. Trustworthy recommendations by filtering fake reviews.

For Businesses: Better insights into product performance and customer feedback. Enhanced strategies based on detailed trend analyses.

Project Phases

Phase 1: Data collection and preprocessing.

Phase 2: Develop and train sentiment analysis models.

Phase 3: Build summarization and spam detection features.

Phase 4: Create dashboards and visualizations.

Phase 5: Testing, validation, and deployment

CODE:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
df = pd.read_csv("/content/Reviews.csv", quotechar='"')
df2=df.copy()
df.head(1000)
df.shape
df.info()
df['Text']

df.drop(['Id','ProductId','UserId','ProfileName','HelpfulnessDenominator','HelpfulnessNumerator','Time',
         'Summary'],axis=1,inplace=True)

df.columns
df.isna().sum()
df.duplicated().sum()
df.drop_duplicates(inplace=True)
df.shape
df['Score'].nunique()
df['Score'].unique()
df['Score'].value_counts()/len(df)*100
plt.figure(figsize=(10,5))
ax=sns.countplot(x=df['Score'])
total=float(len(df))
for p in ax.patches:
    height = p.get_height()
    ax.text(p.get_x()+p.get_width()/2.,height + 75,'{:1.1f} %'.format((height/total)*100), ha="center",
            bbox=dict(facecolor='none', edgecolor='black', boxstyle='round', linewidth=0.5))
ax.set_title('Score Distribution', fontsize=20, y=1.05)
sns.despine(right=True)
sns.despine(offset=5, trim=True)
score_values=df['Score'].value_counts()
plt.pie(score_values,labels=score_values.index)
```

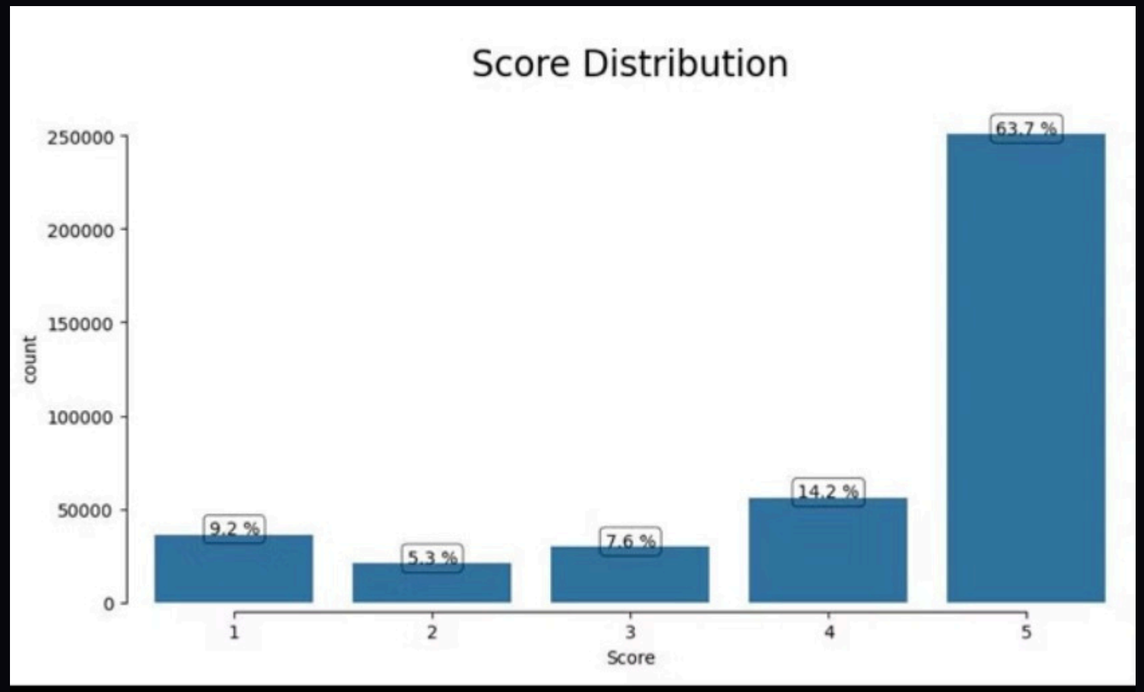
```
plt.pie(score_values,labels=score_values.index)
plt.title('Pie Score Disribution')
plt.show()
new_df= df.groupby('Score').apply(lambda x: x.sample(10000)).reset_index(drop=True)
new_df
new_df.shape
new_df['Score'].value_counts()
plt.figure(figsize=(10,5))
ax=sns.countplot(x=new_df['Score'])
total=float(len(df))
for p in ax.patches:
    height = p.get_height()
    ax.text(p.get_x()+p.get_width()/2.,height + 75,'{:1.1f} %'.format((height/total)*100), ha="center",
            bbox=dict(facecolor='none', edgecolor='black', boxstyle='round', linewidth=0.5))
ax.set_title('New Score Distribution', fontsize=20, y=1.05)
sns.despine(right=True)
sns.despine(offset=5, trim=True)
import string
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from nltk.stem import PorterStemmer
import nltk
nltk.download('stopwords')
stop_words=set(stopwords.words('english'))

stemming=PorterStemmer()
def clean_text(text):

    txt=text.lower()


    tokens=word_tokenize(text)


    tokens=[word for word in tokens if word not in string.punctuation]
```

```
    tokens=[word for word in tokens if word not in string.punctuation]


    tokens=[word for word in tokens if word not in stop_words]


    tokens=[word for word in tokens if not word.isdigit()]


    tokens=[stemming.stem(word) for word in tokens]
    return ' '.join(tokens)
import nltk
nltk.download('punkt_tab')
new_df['cleaned_text'] = new_df['Text'].apply(clean_text)
new_df
!pip install wordcloud
from wordcloud import WordCloud

all_text=' '.join(new_df['cleaned_text'])




wordcloud = WordCloud(width=800, height=400, max_words=1000, background_color='white').generate(all_text)


plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
len(set(all_text))
len(all_text)
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer, TfidfTransformer
from sklearn.linear_model import LogisticRegression
```

```python
from sklearn.linear_model import LogisticRegression

from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
new_df['Score'] = new_df['Score'].apply(lambda x: 1 if x >=3 else 0)
from sklearn.model_selection import train_test_split
X=new_df['cleaned_text']
y=new_df['Score']
X_train, X_test, y_train, y_test = train_test_split(X,y,random_state = 42,test_size = 0.20)

logistic_pipe=Pipeline(
    [
      ('vec',CountVectorizer(stop_words= "english")),
        ('Tf_idf',TfidfTransformer()),
        ('log_rg',LogisticRegression()),

    ]
)
log_fit = logistic_pipe.fit(X_train,y_train)
log_pred=logistic_pipe.predict(X_test)
print('Training accuracy:', log_fit.score(X_train,y_train))
print('Test accuracy:', log_fit.score(X_test,y_test))
sns.heatmap(confusion_matrix(y_test,log_pred), annot=True, fmt="d")
reviews=['This is an amazing product,I will definetly buy it ',
         'very bad,I dont recommend it at all',
'we received this coffee yesterday, and have to say its amazing',
         'experience was terrible',
         'I will buy again from this site,everything was perfect']
prediction=logistic_pipe.predict(reviews)
sentiment=["Positive" if i == 1 else "Negative" for i in prediction]

print(sentiment)
naive_bayes_pipeline = Pipeline([
    ('vec', CountVectorizer(stop_words='english')),
    ('tfidf', TfidfTransformer()),
```

```
prediction=logistic_pipe.predict(reviews)
sentiment=["Positive" if i == 1 else "Negative" for i in prediction]

print(sentiment)
naive_bayes_pipeline = Pipeline([
    ('vec', CountVectorizer(stop_words='english')),
    ('tfidf', TfidfTransformer()),
    ('classifier', MultinomialNB())
])


nb_model = naive_bayes_pipeline.fit(X_train, y_train)
y_pred_nb = naive_bayes_pipeline.predict(X_test)
print(classification_report(y_test, y_pred_nb,digits=4))
sns.heatmap(confusion_matrix(y_test,y_pred_nb), annot=True, fmt="d")
print('Training accuracy of Navie Bayes : ', nb_model.score(X_train,y_train))
print('Test accuracy of Navie Bayes : ', nb_model.score(X_test,y_test))
```

OUTPUTS:




```
[34]
    import nltk
    nltk.download('stopwords')

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.

True
```

```
import nltk
nltk.download('punkt_tab')
```

Python

```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.

True
```

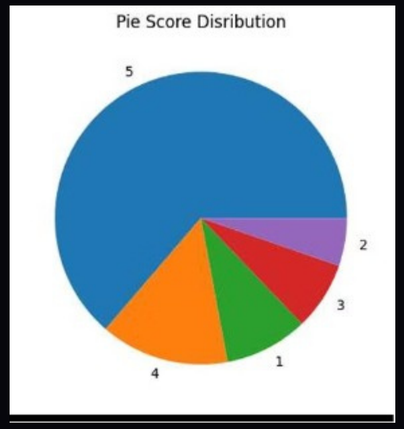
```
new_df['cleaned_text'] = new_df['Text'].apply(clean_text)
```

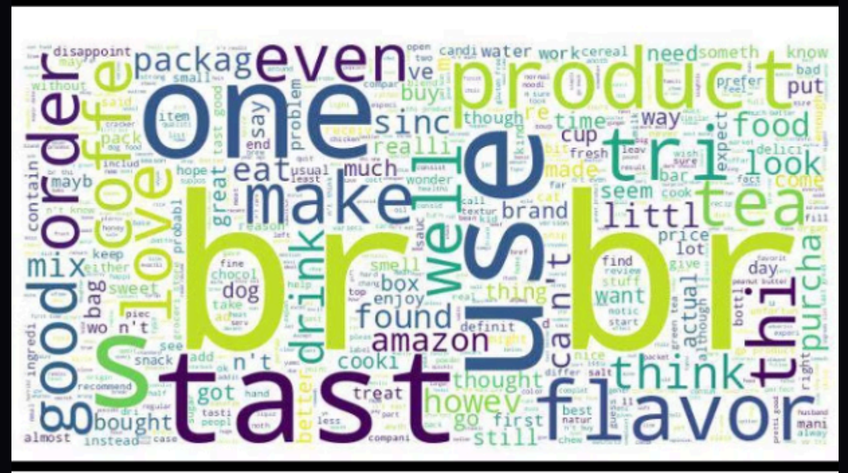
Python

```
new_df
```

Python

	Score	Text	cleaned_text
0	1	I bought and installed two of these traps 8 da...	i bought instal two trap day ago let report re...
1	1	I was looking forward to this product, as my w...	i look forward product wife i like earl grey t...
2	1	Since this type of yeast is no longer availabl...	sinc type yeast longer avail local groceri sto...
3	1	Let me be clear: I wasn't expecting this to ta...	let clear i n't expect tast like top-shelf mar...
4	1	I'm a big fan of Twinings, so when I saw that ...	i 'm big fan twine i saw sold green tea i jump...
...	...	...	...
49995	5	I had never tried 5-Hour Energy before, but ha...	i never tri 5-hour energi heard good thing i r...
49996	5	Delicious organic tomato soup. All of ours wer...	delici organ tomato soup all half soup half no...
49997	5	I couldn't count the number of bottles of Pick...	i could n't count number bottl pickapeppa orig...
49998	5	You open the packaging and immediately smell d...	you open packag immedi smell delici fresh bake...
49999	5	We LOVE Enjoy Life cookies. The Snickerdoodle...	we love enjoy life cooki the snickerdoodl wond...






```
print('Training accuracy:', log_fit.score(X_train,y_train))
print('Test accuracy:', log_fit.score(X_test,y_test))
```

```
Training accuracy: 0.82995
Test accuracy: 0.7913
```

```
sns.heatmap(confusion_matrix(y_test,log_pred), annot=True, fmt="d")
```

C: > Users > adull > AppData > Local > Microsoft > Windows > INetCache > IE > 4NH3YG7P > amazon_reviews[1].ipynb > imp

+ Code + Markdown ···

```
print('Training accuracy:', log_fit.score(X_train,y_train))
print('Test accuracy:', log_fit.score(X_test,y_test))
```

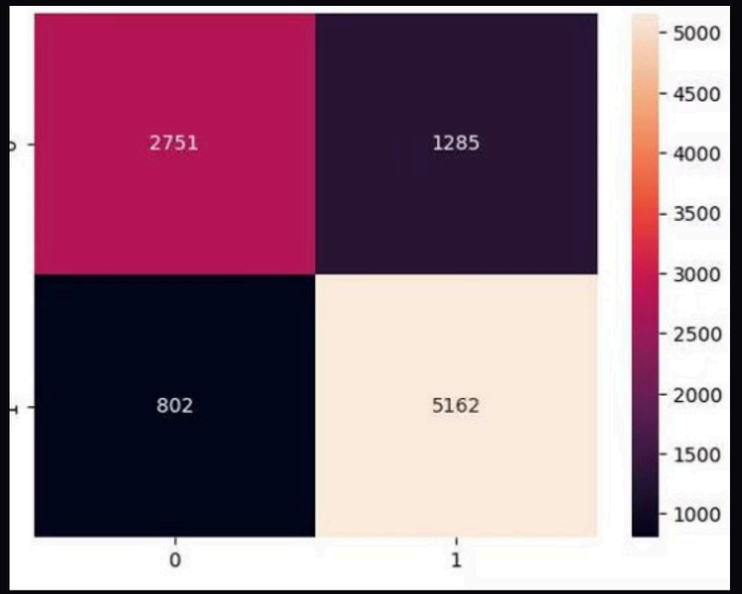
[94]

```
Training accuracy: 0.82995
Test accuracy: 0.7913
```

```
sns.heatmap(confusion_matrix(y_test,log_pred), annot=True, fmt="d")
```

[95]

```
<Axes: >
```



C: > Users > adull > AppData > Local > Microsoft > Windows > INetCache > IE > 4NH3YG7P > amazon_reviews[1].ipynb >

+ Code + Markdown ···

```
reviews=['This is an amazing product,I will definetly buy it ',
         'very bad,I dont recommend it at all',
'we received this coffee yesterday, and have to say its amazing',
         'experience was terrible',
         'I will buy again from this site,everything was perfect']
prediction=logistic_pipe.predict(reviews)
sentiment=["Positive" if i == 1 else "Negative" for i in prediction]
# Predicted : P,N,P,N,P    2 errors
print(sentiment)
```

[56]

```
['Negative', 'Negative', 'Positive', 'Negative', 'Positive']
```

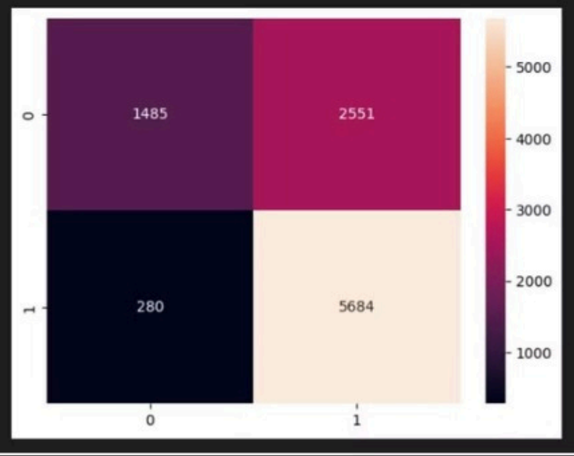
C: > Users > adull > AppData > Local > Microsoft > Windows > INetCache > IE > 4NH3YG7P > amazon_reviews[1].ipynb >

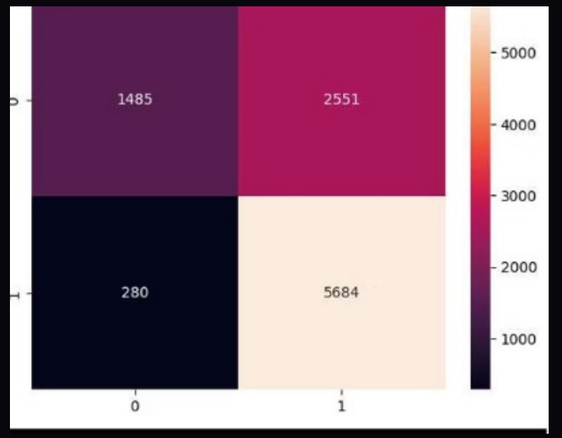
+ Code + Markdown ···

```
···              precision    recall  f1-score   support

           0       0.8414    0.3679    0.5120      4036
           1       0.6902    0.9531    0.8006      5964

    accuracy                         0.7169     10000
   macro avg       0.7658    0.6605    0.6563     10000
weighted avg       0.7512    0.7169    0.6841     10000
```

··· <Axes: >

···





INDIVIDUAL CONTRIBUTIONS FOR THE PROJECT:

1.Design and Development Team

Sai Venkata Pranav Kausugurthi – Develops sentiment analysis models.

Boda Lokesh – Implements feature extraction for key review aspects.

Tanakala Vignesh Venkata Harsha – Designs a user-friendly interface for review insights.

Dasari Govardhan – Optimizes backend systems for large datasets.

Kandala Nishant Reddy – Creates automated review scraping tools.

Racharla Anvesh – Develops APIs for seamless system integration.

2.Data Collection and Database Management Team

Lokeshwara Vikramaditya Chodavarapu – Automates review data collection.

Mallepally Rishivarun – Designs and maintains the database schema.

Tanuj Patnaik Manapuram – Implements data cleaning and preprocessing.

Jarpla Praneeth – Manages large-scale data storage solutions.

Thella Chetan Tanmai – Ensures data security and backup.

3.Documentation and Presentation Team

Vadlamudi Pallavi Naidu – Prepares comprehensive project documentation.

Nainika Challa – Develops visual presentations for the project.

Bondugula Rishi – Creates detailed progress reports.

Adulla Purvi Reddy – Designs impactful presentation slides.

Madhav Banothu – Prepares case studies and examples.

Pedagandham Srivizna – Summarizes user feedback for improvements.

4.Advertisement and Publicity Team

Bukya Shireesh Nayak – Develops promotional materials.

Gajjula Saisreeja – Creates content for blogs and newsletters.

Desam Chandini Lakshmi – Manages stakeholder outreach.

Dinesh Kumar Reddy Guntaka – Tracks and analyzes advertising impact.

THANKYOU