

# 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/data>  
<https://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

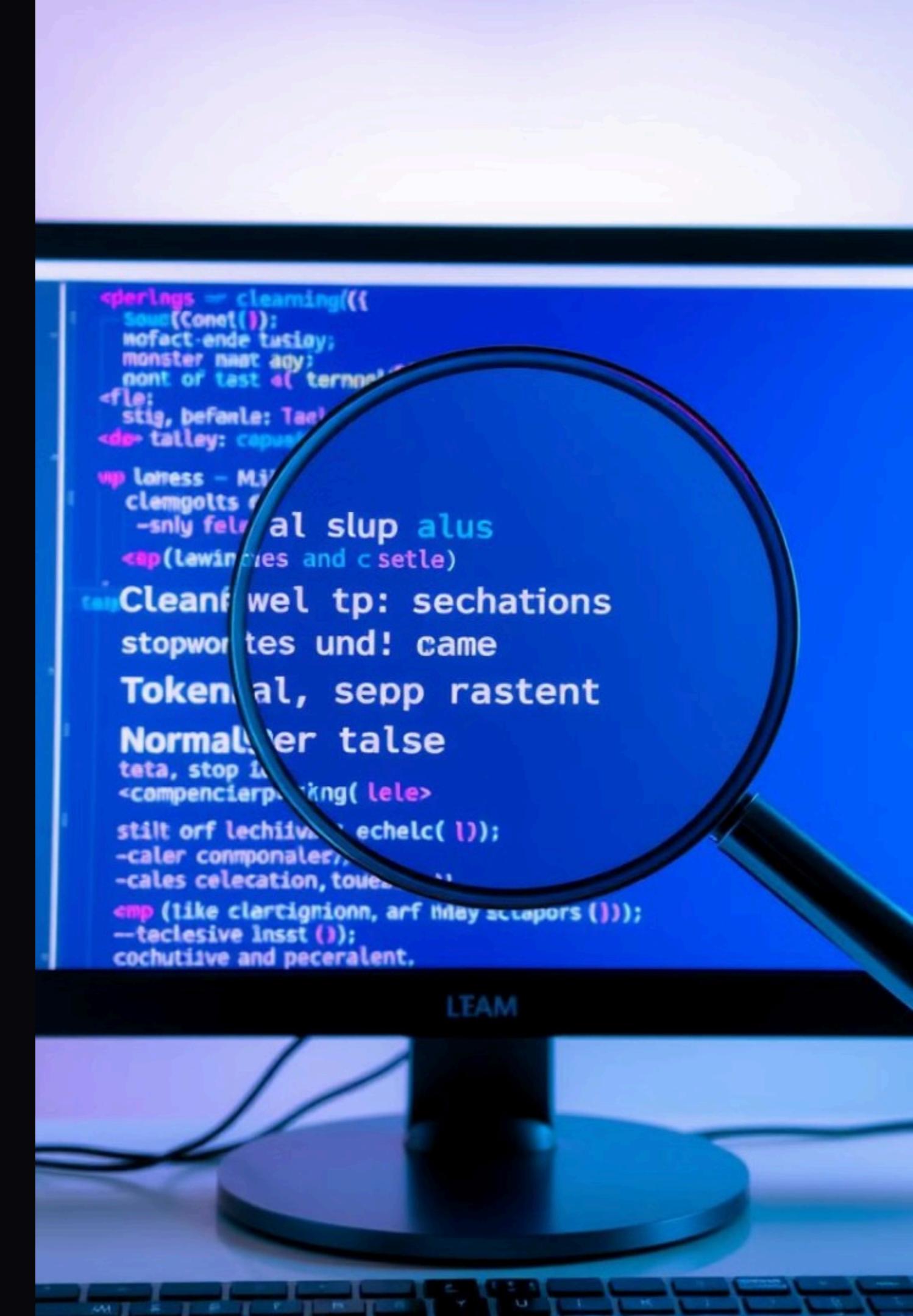
Break reviews into words or phrases for analysis.

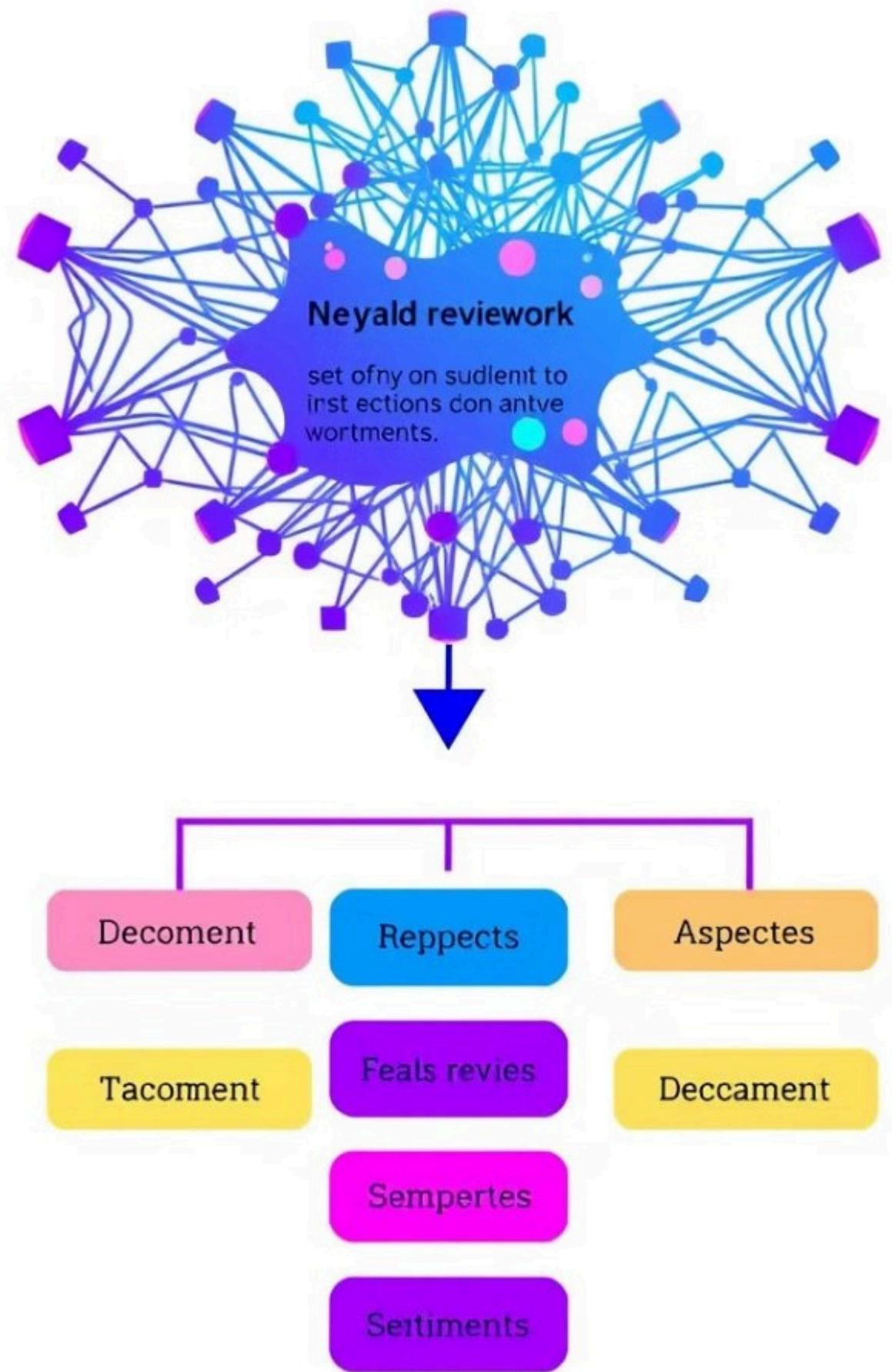
## Stopword Removal

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.

- ↳ recouth's term increcune basting, text.
- ↳ recotured techcupe to chreve bott sask ext.
- ↳ thectune suature fating bo lo.



# 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:

```
1 import pandas as pd
2 import numpy as np
3 import seaborn as sns
4 import matplotlib.pyplot as plt
5 import warnings
6 warnings.filterwarnings('ignore')
7 df = pd.read_csv("/content/Reviews.csv", quotechar='''')
8 df2=df.copy()
9 df.head(1000)
10 df.shape
11 df.info()
12 df['Text']
13
14 df.drop(['Id','ProductId','UserId','ProfileName','HelpfulnessDenominator','HelpfulnessNumerator','Time',
15         'Summary'],axis=1,inplace=True)
16
17 df.columns
18 df.isna().sum()
19 df.duplicated().sum()
20 df.drop_duplicates(inplace=True)
21 df.shape
22 df['Score'].nunique()
23 df['Score'].unique()
24 df['Score'].value_counts()/len(df)*100
25 plt.figure(figsize=(10,5))
26 ax=sns.countplot(x=df['Score'])
27 total=float(len(df))
28 for p in ax.patches:
29     height = p.get_height()
30     ax.text(p.get_x()+p.get_width()/2.,height + 75,'{:1.1f} %'.format((height/total)*100), ha="center",
31             bbox=dict(facecolor='none', edgecolor='black', boxstyle='round', linewidth=0.5))
32 ax.set_title('Score Distribution', fontsize=20, y=1.05)
33 sns.despine(right=True)
34 sns.despine(offset=5, trim=True)
35 score_values=df['Score'].value_counts()
36 plt.pie(score_values,labels=score_values.index)
```

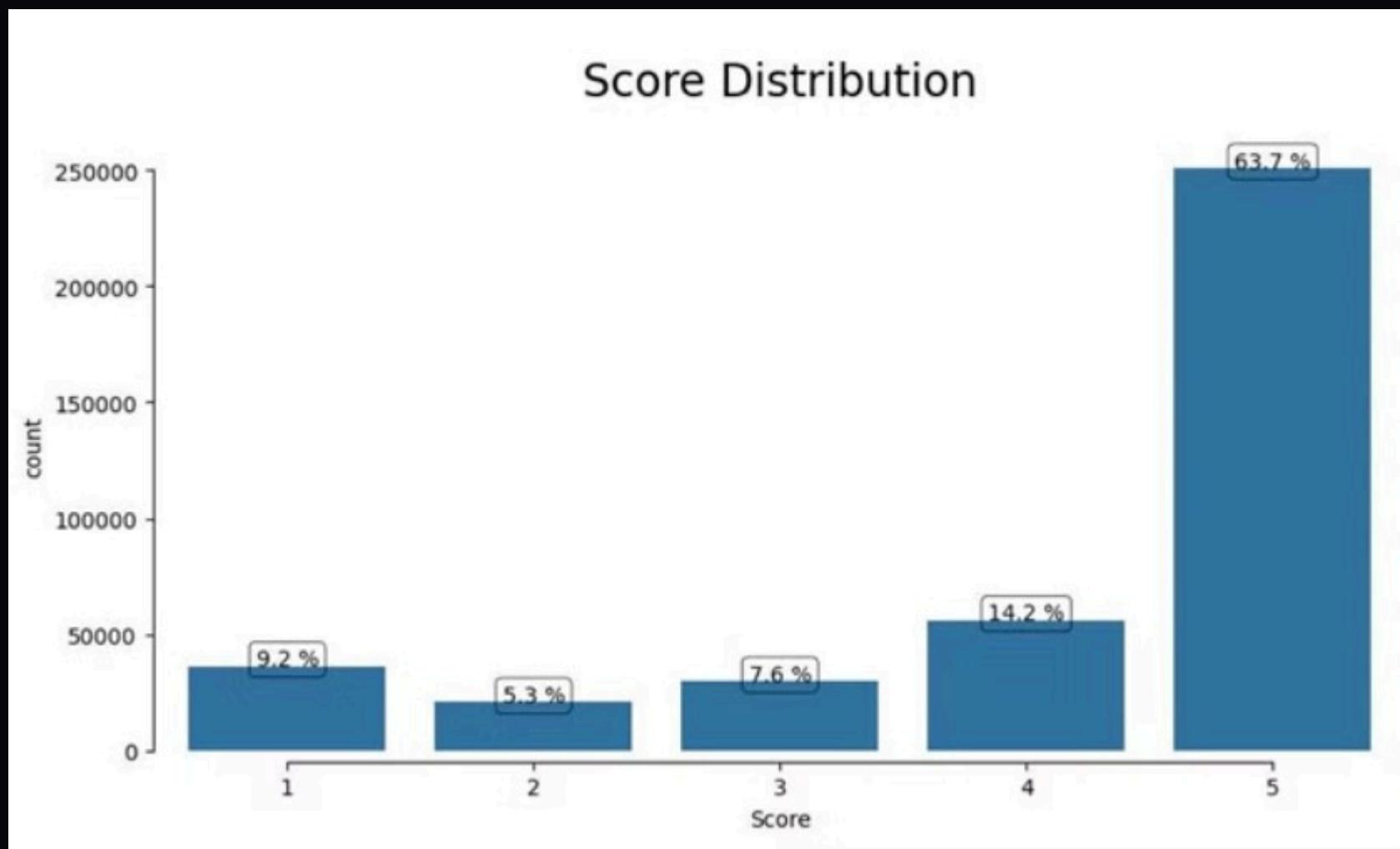
```
36 plt.pie(score_values,labels=score_values.index)
37 plt.title('Pie Score Disribution')
38 plt.show()
39 new_df= df.groupby('Score').apply(lambda x: x.sample(10000)).reset_index(drop=True)
40 new_df
41 new_df.shape
42 new_df['Score'].value_counts()
43 plt.figure(figsize=(10,5))
44 ax=sns.countplot(x=new_df['Score'])
45 total=float(len(df))
46 for p in ax.patches:
47     height = p.get_height()
48     ax.text(p.get_x()+p.get_width()/2.,height + 75,'{:1.1f} %'.format((height/total)*100), ha="center",
49             bbox=dict(facecolor='none', edgecolor='black', boxstyle='round', linewidth=0.5))
50 ax.set_title('New Score Distribution', fontsize=20, y=1.05)
51 sns.despine(right=True)
52 sns.despine(offset=5, trim=True)
53 import string
54 from nltk.tokenize import word_tokenize
55 from nltk.corpus import stopwords
56 from nltk.stem import WordNetLemmatizer
57 from nltk.stem import PorterStemmer
58 import nltk
59 nltk.download('stopwords')
60 stop_words=set(stopwords.words('english'))
61
62 stemming=PorterStemmer()
63 def clean_text(text):
64
65     txt=text.lower()
66
67     tokens=word_tokenize(text)
68
69
70     tokens=[word for word in tokens if word not in string.punctuation]
```



```
105 from sklearn.linear_model import LogisticRegression
106
107 from sklearn.svm import SVC
108 from sklearn.naive_bayes import MultinomialNB
109 from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
110 new_df['Score'] = new_df['Score'].apply(lambda x: 1 if x >=3 else 0)
111 from sklearn.model_selection import train_test_split
112 X=new_df['cleaned_text']
113 y=new_df['Score']
114 X_train, X_test, y_train, y_test = train_test_split(X,y,random_state = 42,test_size = 0.20)
115
116 logistic_pipe=Pipeline(
117     [
118         ('vec',CountVectorizer(stop_words= "english")),
119         ('Tf_idf',TfidfTransformer()),
120         ('log_rg',LogisticRegression()),
121     ]
122 )
123
124 log_fit = logistic_pipe.fit(X_train,y_train)
125 log_pred=logistic_pipe.predict(X_test)
126 print('Training accuracy:', log_fit.score(X_train,y_train))
127 print('Test accuracy:', log_fit.score(X_test,y_test))
128 sns.heatmap(confusion_matrix(y_test,log_pred), annot=True, fmt="d")
129 reviews=['This is an amazing product,I will definetly buy it ',
130           'very bad,I dont recommend it at all',
131           'we received this coffee yesterday, and have to say its amazing',
132           'experience was terrible',
133           'I will buy again from this site,everything was perfect']
134 prediction=logistic_pipe.predict(reviews)
135 sentiment=[ "Positive" if i == 1 else "Negative" for i in prediction]
136
137 print(sentiment)
138 naive_bayes_pipeline = Pipeline([
139     ('vec', CountVectorizer(stop_words='english')),
140     ('tfidf', TfidfTransformer()),
```

```
134 prediction=logistic_pipe.predict(reviews)
135 sentiment=["Positive" if i == 1 else "Negative" for i in prediction]
136
137 print(sentiment)
138 naive_bayes_pipeline = Pipeline([
139     ('vec', CountVectorizer(stop_words='english')),
140     ('tfidf', TfidfTransformer()),
141     ('classifier', MultinomialNB())
142 ])
143
144
145 nb_model = naive_bayes_pipeline.fit(X_train, y_train)
146 y_pred_nb = naive_bayes_pipeline.predict(X_test)
147 print(classification_report(y_test, y_pred_nb,digits=4))
148 sns.heatmap(confusion_matrix(y_test,y_pred_nb), annot=True, fmt="d")
149 print('Training accuracy of Navie Bayes : ', nb_model.score(X_train,y_train))
150 print('Test accuracy of Navie Bayes : ', nb_model.score(X_test,y_test))
151
152
153
154
155
156
157
158
159
160
161
162
163
```

# OUTPUTS:

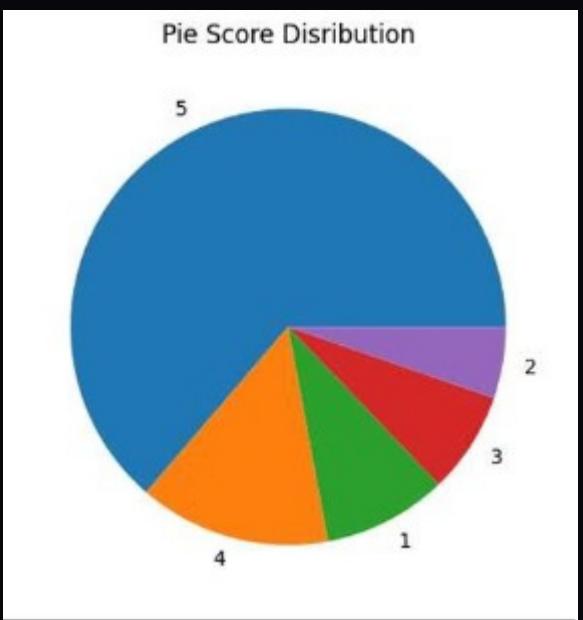


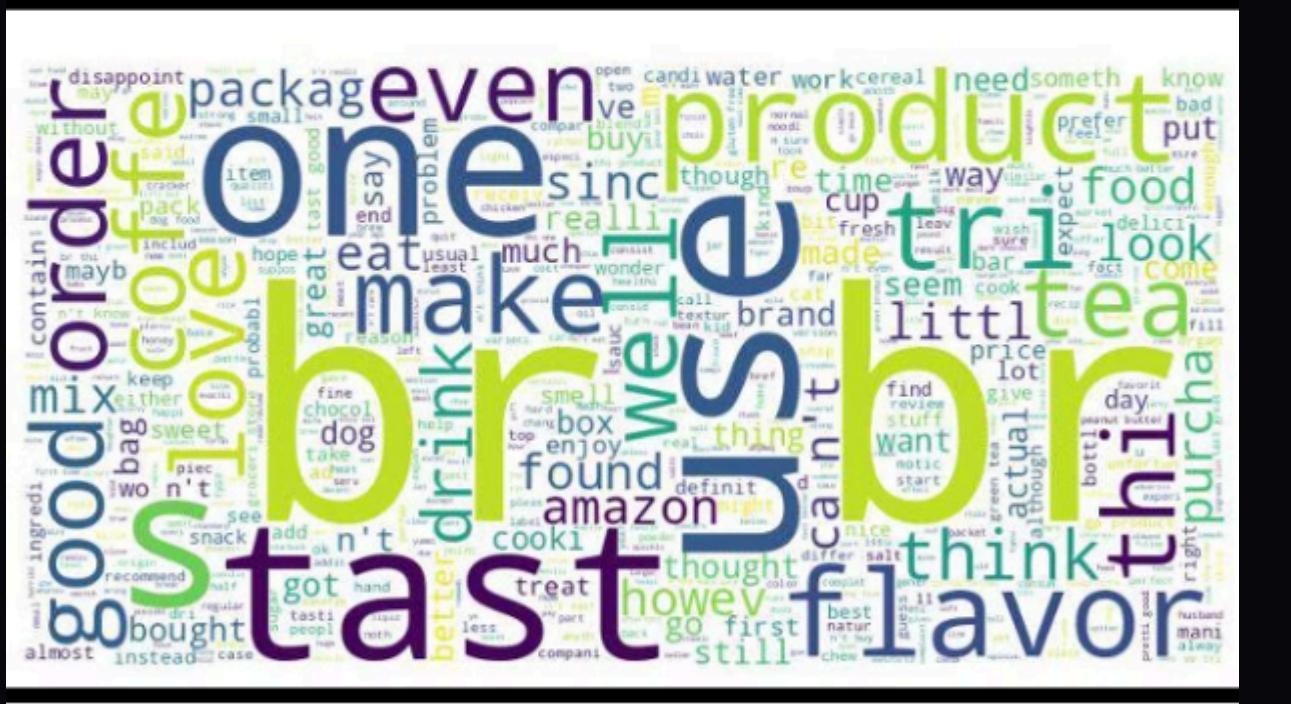
```
▷ import nltk
nltk.download('stopwords')
[34]
...
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]  Unzipping corpora/stopwords.zip.
...
... True
```

```
[1]: In [1]: import nltk  
[1]: nltk.download('punkt_tab')  
[1]:  
... [nltk_data] Downloading package punkt_tab to /root/nltk_data...  
[nltk_data]  unzipping tokenizers/punkt_tab.zip.  
...  
True  
  
D> new_df['cleaned_text'] = new_df['Text'].apply(clean_text)  
[2]:  
  
In [3]: new_df  
[3]:  
...  


|       | 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-shell 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... |


```





```
[53]: 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\4NHQYQ7P> 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))
```

[54]

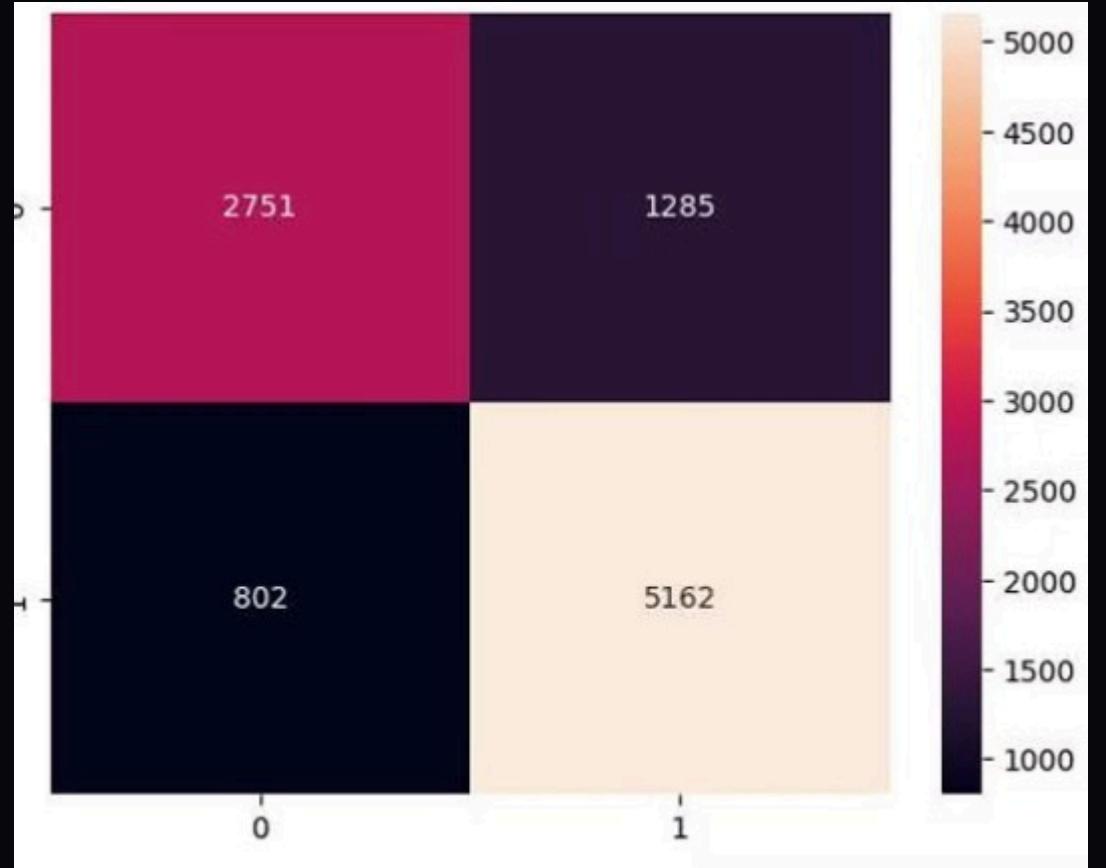
```
... Training accuracy: 0.82995
test accuracy: 0.7913
```

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

[55]

```
<Axes: >
```

[56]



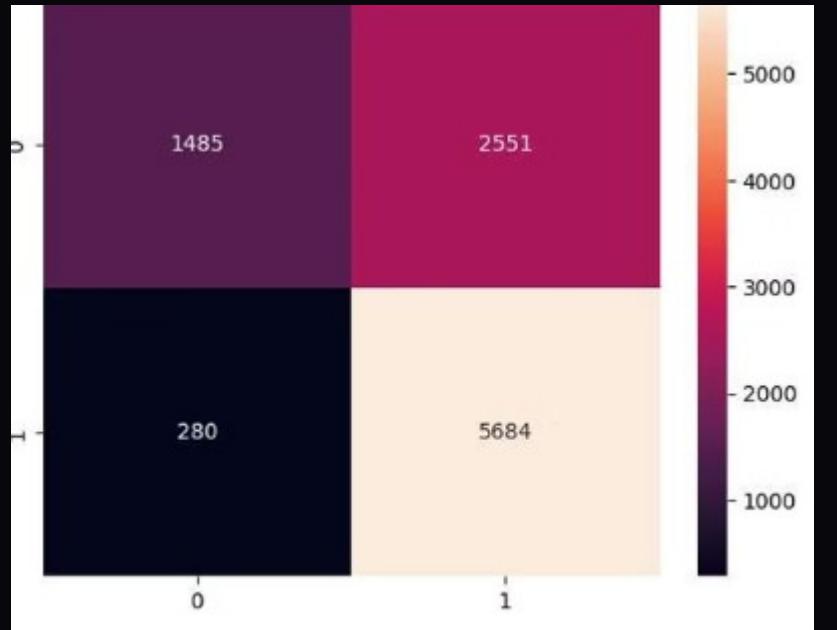
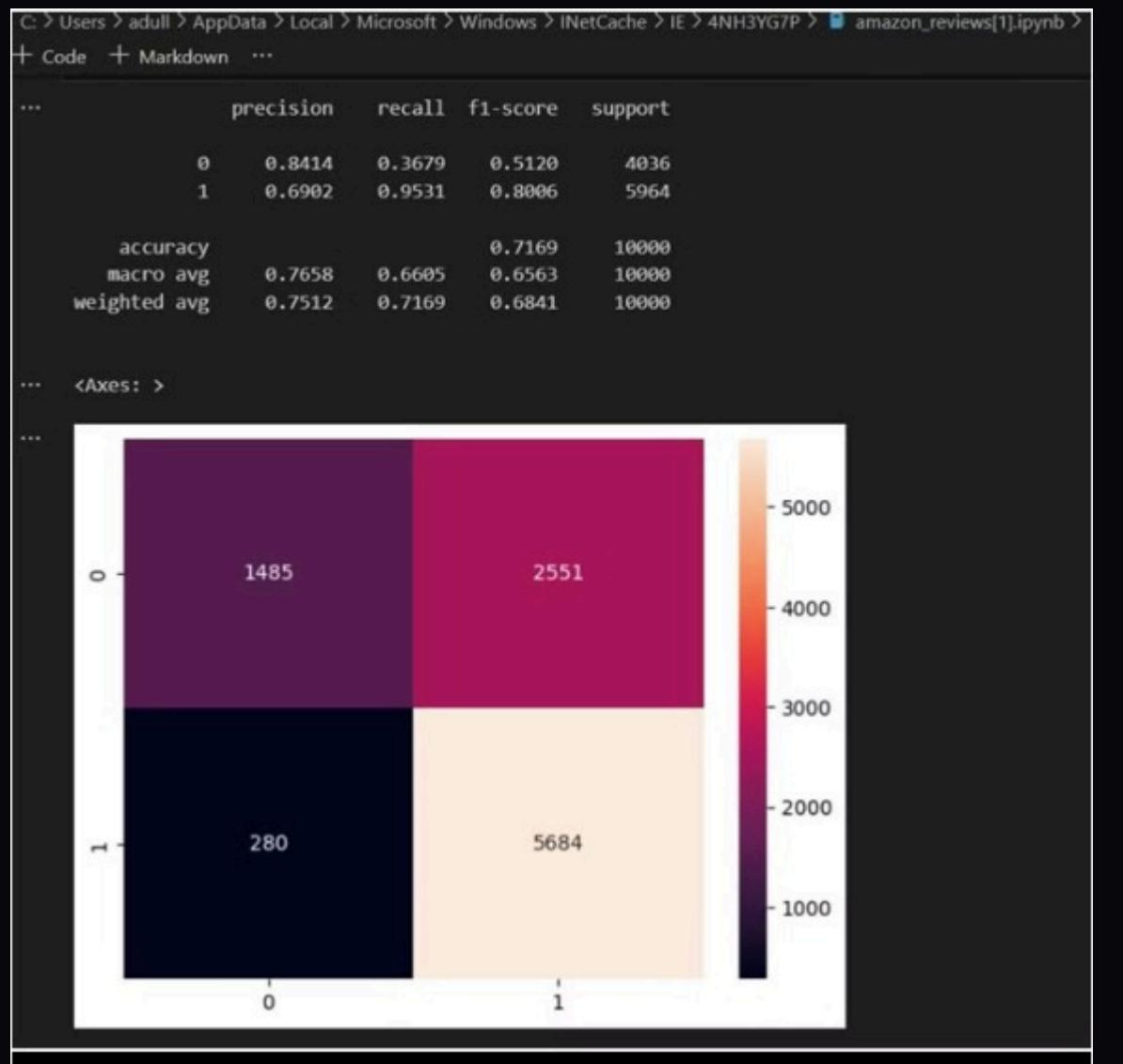
C: > Users > adull > AppData > Local > Microsoft > Windows > INetCache > IE > 4NH3YG7P > amazon\_reviews[1].ipynb >

+ 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']
```



# INDIVIDUAL CONTRIBUTIONS FOR THE PROJECT:

## 1. Design and Development Team

### **Sai Venkata Pranav Kausugurthi | Boda Lokesh | Tanakala Vignesh Venkata Harsha**

Created the main system to analyze and classify customer reviews. Designed a user-friendly interface and added visual tools like graphs and word clouds

### **Dasari Govardhan | Kandala Nishant Reddy | Racharla Anvesh**

Improved system performance to handle large amounts of data quickly. Built APIs to connect different parts of the system. Focused on using efficient machine learning methods

## 2.Data Collection and Database Management Team

**Lokeshwara Vikramaditya Chodavarapu | Mallepally Rishivarun | Tanuj Patnaik Manapuram**

Collected reviews automatically from Amazon and ensured the data was accurate. Designed and managed a database to store and access reviews easily.

**Jarpla Praneeth | Thella Chetan Tanmai**

Cleaned and organized the data to keep it consistent. Set up systems to store large amounts of data and improve speed. Took care of data security and backups.

### 3. Documentation and Presentation Team

**Vadlamudi Pallavi Naidu | Nainika Challa | Bonugula Rishi**

Wrote detailed documents explaining how the system works. Created simple and clear presentations to show the project's progress and results.

**Adulla Purvi Reddy | Madhav Banothu | Pedagandham Srivizna**

Prepared user guides and reports. Collected user feedback and suggested improvements. Designed slides for the final presentation.

## 4. Advertisement and Publicity Team

**Bukya Shireesh Nayak | Gajjula Saisreeja**

Made promotional content like posters, videos, and social media posts. Managed the project's online presence through blogs and newsletters.

**Desam Chandini Lakshmi | Dinesh Kumar Reddy Guntaka**

Managing GitHub repository .Checked how well advertising worked and suggested improvements.

THANK YOU