

PROJECT: SENTIMENT ANALYSIS FOR MARKETING



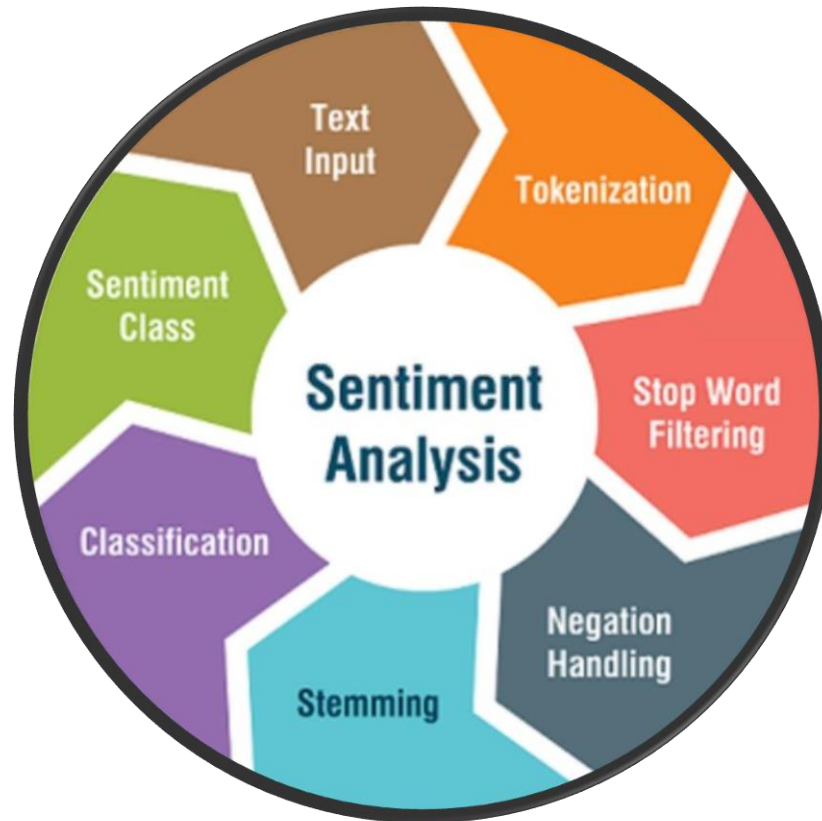
PRESENTED BY

S. SAHANA

821021104038

PHASE 3 : DEVELOPMENT PART

Start building the Sentiment Analysis for Marketing to analysis customers sentiments for competitor products.



Sentiment analysis is a marketing tool that helps you examine the way people interact with a brand online. This method is more comprehensive than traditional online marketing tracking, which measures the number of online interactions that customers have with a brand, like comments and shares.

APPLICABILITY:

AI powered to enhance products by understanding customers likes and dislikes.

TABLE OF CONTENTS:

- Introduction
- Training AI Models for Sentiment Analysis
- Importing Essential Libraries
- Necessary Steps to Follows
- Import Datasets

➤ INTRODUCTION:

- Sentiment Analysis, often referred to as opinion mining, is a powerful technique within the field of Natural Language Processing (NLP).
- At its core, sentiment analysis involves teaching machines to understand and interpret human emotions and opinions expressed within text data.
- By analyzing the sentiment behind words and phrases, AI models can classify text as positive, negative, or neutral, thus providing valuable insights into people's attitudes, feelings, and reactions.

TRAINING AI MODELS FOR SENTIMENT ANALYSIS:

Training AI models for sentiment analysis involves these steps:

➤ **Data Collection:**

Gather a labeled dataset with text samples and sentiment labels (positive negative).

➤ **Text Preprocessing:**

Clean text by removing punctuation, special characters, and lowercase conversion.

➤ **Tokenization:**

Break text into smaller units (tokens) like words.

➤ **Feature Extraction:**

Convert tokens into numerical representations using techniques like TF-IDF.

➤ **Model Selection:**

Choose an algorithm like Naive Bayes, SVM, or RNN.

➤ **Model Training:**

Train the model on labeled data to learn sentiment patterns.

➤ **Model Evaluation:**

Measure model performance with metrics like accuracy and precision.

➤ **Deployment:**

Deploy the model to predict sentiment in new text data.

IMPORTING ESSENTIAL LIBRARIES:

Data Analysis and Visualization Libraries:

❖ Pandas

- ❖ NumPy
- ❖ Seaborn and Matplotlib

Text Preprocessing Libraries:

- ❖ NLTK (Natural Language Toolkit)
- ❖ String and Word Cloud
- ❖ TfidfVectorizer

Data Splitting and Model Training Libraries:

- ❖ train_test_split
- ❖ Logistic Regression and MultinomialNB
- ❖ Naive Bayes

Model Evaluation Libraries

- ❖ Metrics and Display Tools
- ❖ Classification reports and confusion matrices.

NECESSARY STEPS TO FOLLOWS:

IN [1]:

Data Analysis and Visualization

```
import pandas as pd
```

```
import numpy as np
```

```
import seaborn as sns
```

```
import matplotlib.pyplot as plt
```

Text Preprocessing

```
import string from nltk.corpus
```

```
import stopwords from nltk. stem import  
PorterStemmer from wordcloud
```

```
import WordCloud from sklearn. feature  
extraction. text
```

```
import TfidfVectorizer
```


Data Splitting and Model Training

```
from sklearn.model_selection import  
train_test_split
```

```
from sklearn.linear_model import  
LogisticRegression
```

```
from sklearn.naive_bayes import  
MultinomialNB
```

#Model Evaluation

```
from sklearn.metrics import (  
accuracy_score,  
precision_score,  
recall_score,  
f1_score,  
classification_report,  
confusion_matrix,  
ConfusionMatrixDisplay)
```

IMPORT DATASETS:

IN [2]:

```
df = pd.read_csv('amazon_reviews.csv')
```

IN [3]:

```
df.head(5)
```

OUTPUT:

	rating	date	variation	verified_reviews	feedback
0	5	31-Jul-18	Charcoal Fabric	Love my Echo!	1
1	5	31-Jul-18	Charcoal Fabric	Loved it!	1
2	4	31-Jul-18	Walnut Finish	Sometimes while playing a game, you can answer...	1
3	5	31-Jul-18	Charcoal Fabric	I have had a lot of fun with this thing. My 4 ...	1
4	5	31-Jul-18	Charcoal Fabric	Music	1

DATA INSPECTION:

IN [4]:

```
df.info()
```

Data Information

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3150 entries, 0 to 3149
Data columns (total 5 columns):
#   Column                Non-Null Count  Dtype
---  -
0   rating                 3150 non-null   int64
1   date                   3150 non-null   object
2   variation               3150 non-null   object
3   verified_reviews       3150 non-null   object
4   feedback               3150 non-null   int64
dtypes: int64(2), object(3)
memory usage: 123.2+ KB
```

Display the first 5 full reviews with a space in between

IN [5]:

```
for index, row in df.head(5).iterrows():
    print(f"Review {index + 1}: {row[
'verified_reviews']}\n")
```

Review 1: Love my Echo!

Review 2: Loved it!

Review 3: Sometimes while playing a game, you can answer a question correctly but Alexa says you got it wrong and answers the same as you. I like being able to turn lights on and off while away from home.

Review 4: I have had a lot of fun with this thing. My 4 yr old learns about dinosaurs, i control the lights and play games like categories. Has nice sound when playing music as well.

Review 5: Music

DATA PREPROCESSING:

IN [6]:

```
null_mask = df.isnull()
null_values = null_mask.sum().sum()
print ("Number of null values:", null_values)
```

EXPLORATORY DATA ANALYSIS:

IN [7]:

```
print("\nSummary Statistics:")
summary_stats = df.describe()
print(summary_stats)
```

Summary Statistics:

	rating	feedback
count	3150.000000	3150.000000
mean	4.463175	0.918413
std	1.068506	0.273778
min	1.000000	0.000000
25%	4.000000	1.000000
50%	5.000000	1.000000
75%	5.000000	1.000000
max	5.000000	1.000000

DISTRIBUTION OF SENTIMENTS:

IN [8]:

```
Print ("\n Distribution of Sentiments:")
```

```
sentiment_counts =
```

```
df['feedback'].value_counts()
```

```
print (sentiment_counts)
```

```
Distribution of Sentiments:  
1      2893  
0       257  
Name: feedback, dtype: int64
```

DISTRIBUTION OF RATINGS:

IN [9]:

```
Print ("\nDistribution of Ratings:")
```

```
rating_counts =
```

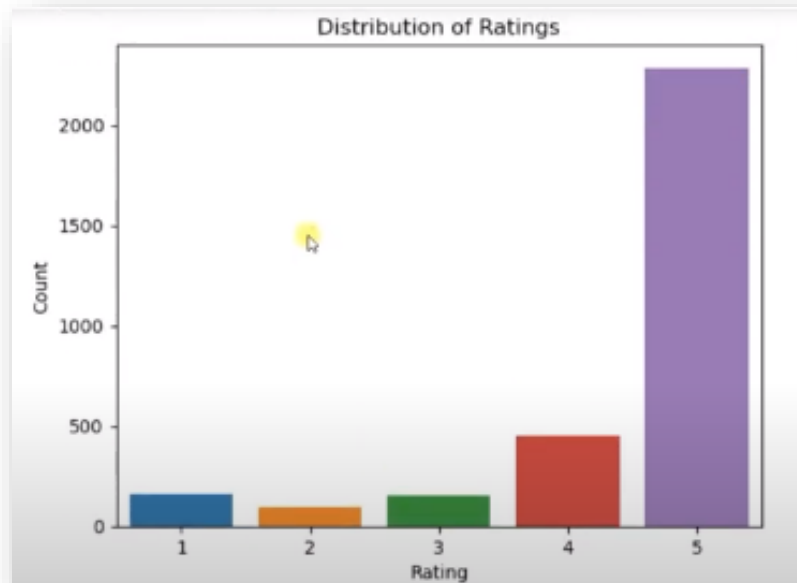
```
df['rating'].value_counts().sort_index()
```

```
print (rating_counts)
```

```
Distribution of Ratings:  
1      161  
2       96  
3      152  
4      455  
5     2286  
Name: rating, dtype: int64
```

IN [9.1]:

```
#plt.figure(figsize=(8, 6))  
sns.countplot (data=df, x='rating')  
plt.title('Distribution of Ratings')  
plt.xlabel('Rating')  
plt.ylabel('Count')  
plt.show()
```



DISTRIBUTION OF VARIATIONS:

IN [10]:

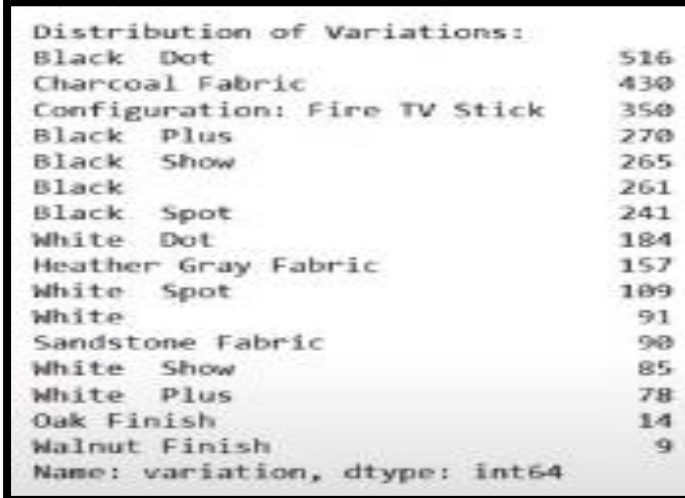
```

Print ("\nDistribution of Variations:")

variation_counts=df["variation"]
.value_counts()

print (variation_counts)

```



```

Distribution of Variations:
Black Dot                    516
Charcoal Fabric              430
Configuration: Fire TV Stick 350
Black Plus                   270
Black Show                   265
Black                        261
Black Spot                   241
White Dot                    184
Heather Gray Fabric          157
White Spot                   109
White                        91
Sandstone Fabric              90
White Show                    85
White Plus                    78
Oak Finish                    14
Walnut Finish                  9
Name: variation, dtype: int64

```

In [10.1]:

```

#plt.figure(figsize=(12, 6))

sns.countplot (data=df, x='variation',
order=df['variation'].value_counts().index)

plt.title('Distribution of Variations')

plt.xlabel('Variation')

plt.ylabel('Count')

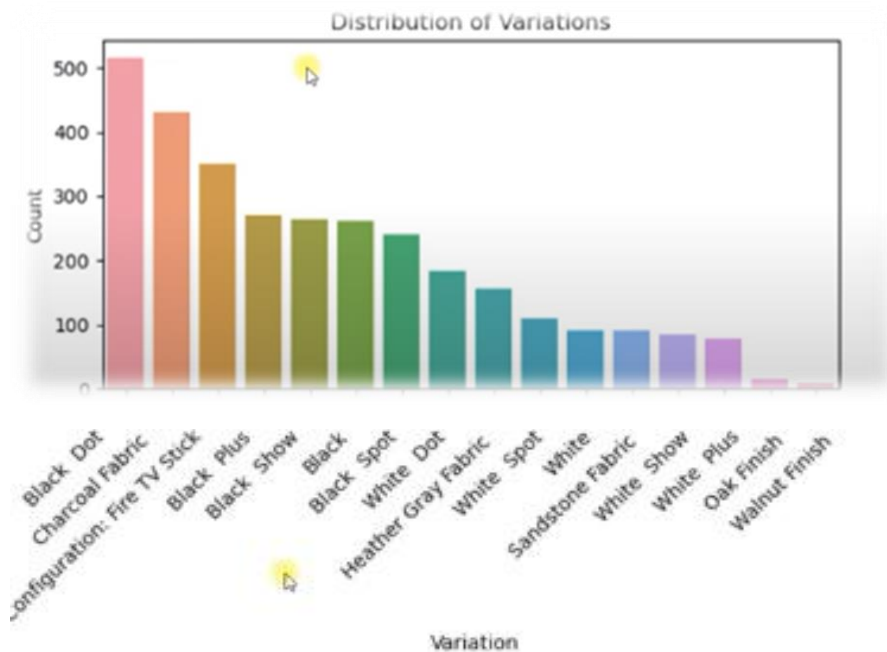
```



```
plt.xticks (rotation=45, ha='right')
```

```
plt.tight_layout()
```

```
plt.show()
```



DISTRIBUTION OF SENTIMENTS:

IN [11]:

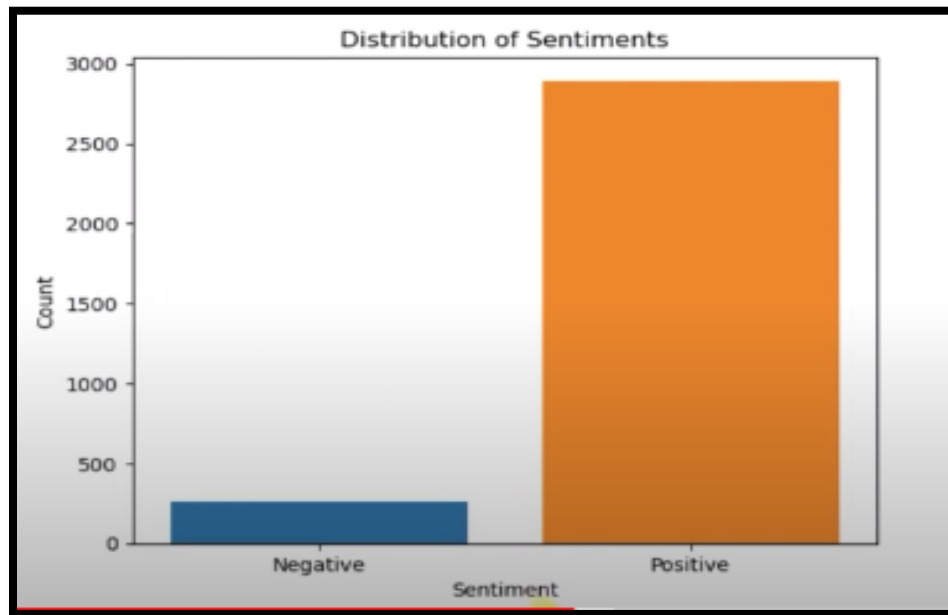
```
#plt.figure(figsize=(8, 6))
```

```
sns.countplot (data=df, x='feedback')
```

```
plt.title ('Distribution of Sentiments')
```

```
plt.xlabel ('Sentiment')  
plt.ylabel ('Count')  
plt.xticks ([0, 1], ['Negative', 'Positive'])  
plt.show()
```

#Distribution of Sentiments (Feedback)



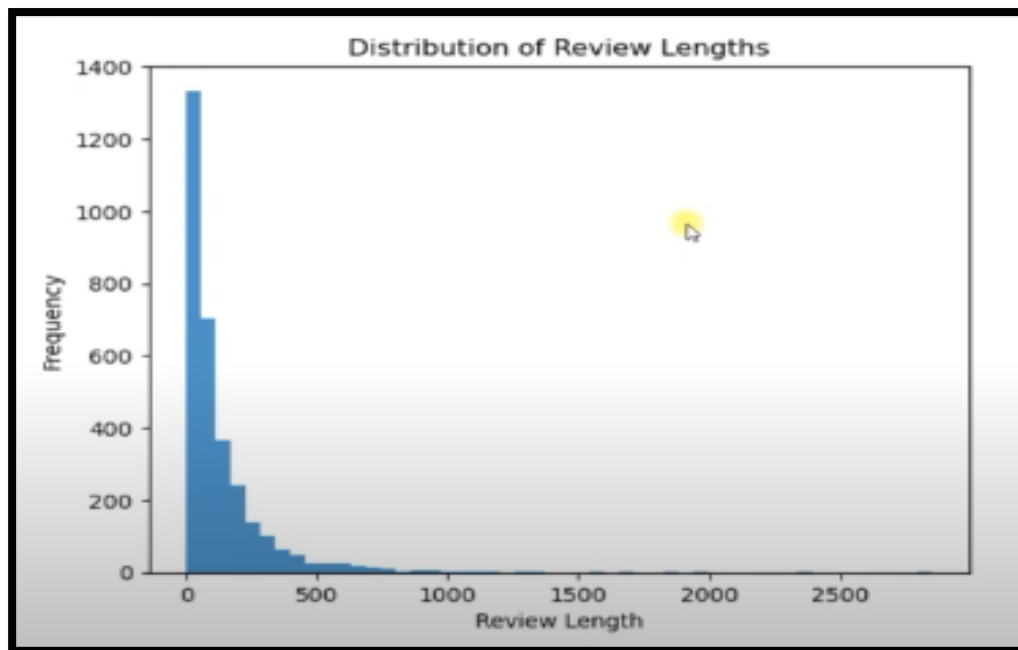
DISTRIBUTION OF REVIEW LENGTHS:

IN [12]:

Calculate the Length of each review

```
df['review_length']=df['verified_reviews'].  
apply(len)
```

```
#plt.figure(figsize=(10, 6))  
plt.hist(df['review_length'],bins=50,  
alpha=0.8)  
plt.xlabel('Review Length')  
plt.ylabel('Frequency')  
plt.title('Distribution of Review Lengths')  
plt.show()
```



TEXT PREPROCESSING :

- Tokenization
- Punctuation Removal and Lowercasing

- Stopword Removal
- Stemming

FEATURE EXTRACTION USING TF-IDF IN SENTIMENT ANALYSIS

- ❖ In sentiment analysis, feature extraction is a crucial step that converts processed text data into numbers, suitable for machine learning.
- ❖ One common technique is TF-IDF (Term Frequency-Inverse Document Frequency), which assigns weights to words in text documents.
- ❖ It measures word importance within a document while considering its frequency across all documents.

MODEL SELECTION AND TRAINING:

❖ The models employed in this project-
Logistic Regression and Multinomial Naive Bayes-and their significance in sentiment analysis.

- Logistic Regression
- Multinomial Naive Bayes

MODEL EVALUATION METRICS:

In the field of sentiment analysis, assessing how well our trained models perform is crucial. We use model evaluation metrics to measure their classification accuracy.

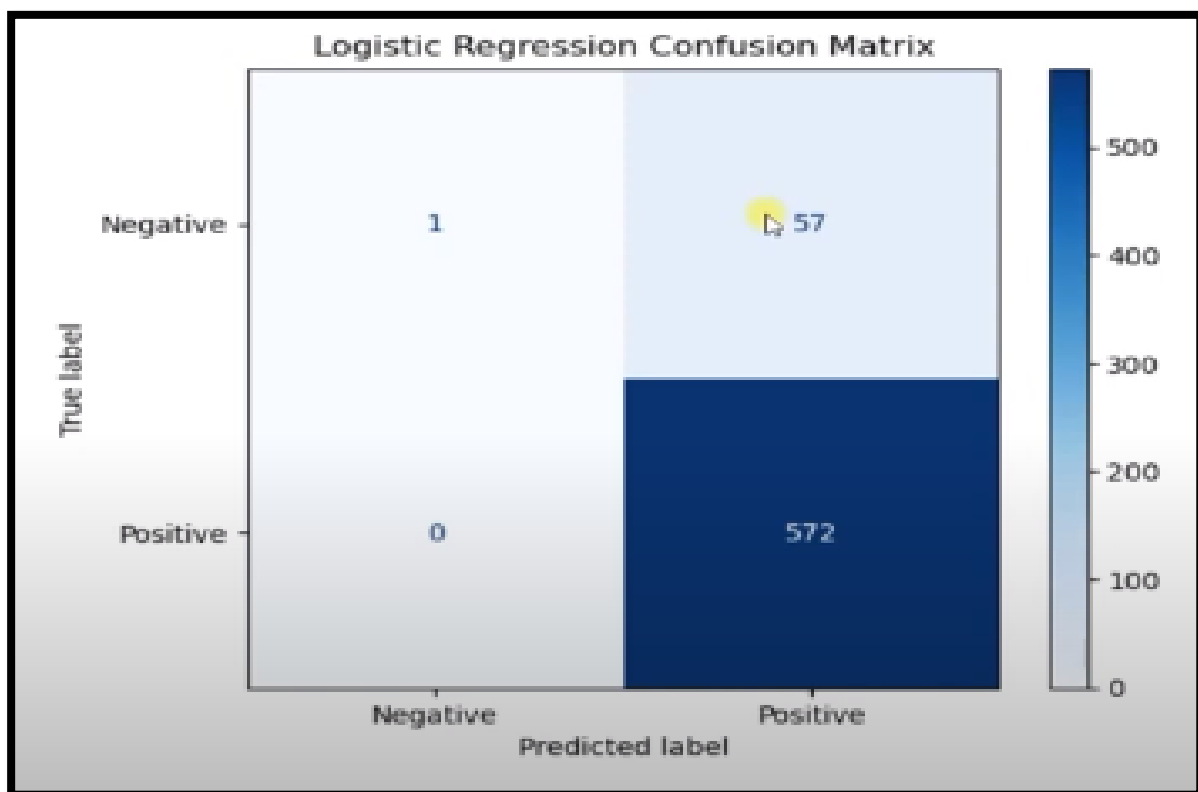
IMPORTANT MODEL EVALUATION METRICS:

- Accuracy
- Precision
- Recall (Sensitivity)

➤ F1-Score

LOGISTIC REGRESSION CONFUSION MATRIX:

`plot_confusion_matrix(logistic_regression
model, X_test, y_test, title='Logistic Regression
Confusion Matrix')`



MULTINOMIAL NAIVE BAYES

CONFUSION MATRIX

`plot_confusion_matrix(multinomial_nb_model, X_test, y_test, title='Multinomial Naive Bayes Confusion Matrix')`

