**### Final Project Summary**

Project Title: Sentiment Analysis and Topic Modeling of Amazon Product Reviews

Objective:

The project focuses on analyzing sentiments expressed in Amazon product reviews and clustering similar reviews using topic modeling techniques. The goal is to predict sentiment accurately and uncover underlying themes within the reviews.

Data Overview:

- Train Data: Includes labeled reviews for model training and evaluation.

- Test Data Hidden: Contains reviews without sentiment labels, used for prediction.

- Test Data: Contains reviews with known sentiments, used to assess model performance.

Key Steps and Implementation:

1. Data Loading and Initial Exploration:

- Loaded datasets and displayed initial rows to understand the structure and content of the data.

python

print("Train Data:")

print(train\_data.head())

print("\nTest Data Hidden:")

print(test\_data\_hidden.head())

print("\nTest Data:")

print(test\_data.head())

2. Exploratory Data Analysis (EDA):

- Sentiment Distribution: Plotted the distribution of sentiment categories in the training data to visualize class distribution.

python

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

sns.countplot(data=train\_data, x='sentiment')

plt.title('Distribution of Sentiment Categories')

plt.show()

3. Addressing Class Imbalance:

- Identified and addressed class imbalance by upsampling minority sentiment classes to balance the dataset. Plotted the balanced distribution to verify effectiveness.

python

from sklearn.utils import resample

majority\_class = train\_data[train\_data.sentiment == 'Neutral']

minority\_classes = train\_data[train\_data.sentiment != 'Neutral']

minority\_classes\_upsampled = resample(minority\_classes, replace=True, n\_samples=len(majority\_class), random\_state=123)

train\_data\_balanced = pd.concat([majority\_class, minority\_classes\_upsampled])

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

sns.countplot(data=train\_data\_balanced, x='sentiment')

plt.title('Balanced Distribution of Sentiment Categories')

plt.show()

4. Feature Engineering:

- Applied TF-IDF Vectorizer to transform review text into numerical features for machine learning models.

python

from sklearn.feature\_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(stop\_words='english', max\_features=5000)

X = vectorizer.fit\_transform(train\_data\_balanced['reviews.text'])

y = train\_data\_balanced['sentiment']

5. Model Implementation and Evaluation:

- Multinomial Naive Bayes:

- Trained and evaluated using confusion matrix and classification report.

python

from sklearn.naive\_bayes import MultinomialNB

from sklearn.metrics import classification\_report, confusion\_matrix

nb\_model = MultinomialNB()

nb\_model.fit(X\_train, y\_train)

y\_pred = nb\_model.predict(X\_test)

print("Naive Bayes Model Evaluation:")

print(confusion\_matrix(y\_test, y\_pred))

print(classification\_report(y\_test, y\_pred))

- Support Vector Machine (SVM):

- Trained using a linear kernel and evaluated performance.

python

from sklearn.svm import SVC

svm\_model = SVC(kernel='linear', probability=True)

svm\_model.fit(X\_train, y\_train)

y\_pred\_svm = svm\_model.predict(X\_test)

print("SVM Model Evaluation:")

print(confusion\_matrix(y\_test, y\_pred\_svm))

print(classification\_report(y\_test, y\_pred\_svm))

- Neural Networks (LSTM):

- Implemented an LSTM model using TensorFlow. Trained the model and evaluated its accuracy. Displayed sample predictions.

python

import tensorflow as tf

from tensorflow.keras.preprocessing.text import Tokenizer

from tensorflow.keras.preprocessing.sequence import pad\_sequences

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout

from sklearn.preprocessing import LabelEncoder

label\_encoder = LabelEncoder()

y\_encoded = label\_encoder.fit\_transform(train\_data['sentiment'])

tokenizer = Tokenizer(num\_words=5000)

tokenizer.fit\_on\_texts(train\_data['reviews.text'])

X\_sequences = tokenizer.texts\_to\_sequences(train\_data['reviews.text'])

X\_padded = pad\_sequences(X\_sequences, maxlen=100)

model = Sequential([

Embedding(5000, 128, input\_length=100),

LSTM(128, return\_sequences=True),

Dropout(0.5),

LSTM(64),

Dense(len(label\_encoder.classes\_), activation='softmax')])

model.compile(loss='sparse\_categorical\_crossentropy', optimizer='adam', metrics=['accuracy'])

history = model.fit(X\_train, y\_train, epochs=5, batch\_size=64, validation\_data=(X\_test, y\_test))

loss, accuracy = model.evaluate(X\_test, y\_test)

print(f'LSTM Model Loss: {loss}')

print(f'LSTM Model Accuracy: {accuracy}')

6. Model Optimization:

- Grid Search for SVM: Performed hyperparameter tuning to find the best SVM parameters.

python

from sklearn.model\_selection import GridSearchCV

param\_grid = {

'C': [0.1, 1, 10],

'kernel': ['linear', 'rbf'],

'gamma': ['scale', 'auto']}

grid\_search = GridSearchCV(svm, param\_grid, cv=3, n\_jobs=-1, verbose=2)

grid\_search.fit(X\_train, y\_train)

print(f'Best Parameters: {grid\_search.best\_params\_}')

print(f'Best Score: {grid\_search.best\_score\_}')

7. Topic Modeling:

- Applied Latent Dirichlet Allocation (LDA) to identify and display latent topics within the reviews.

python

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.decomposition import LatentDirichletAllocation

count\_vectorizer = CountVectorizer(stop\_words='english', max\_features=5000)

X\_count = count\_vectorizer.fit\_transform(train\_data\_balanced['reviews.text'])

lda\_model = LatentDirichletAllocation(n\_components=5, random\_state=42)

lda\_topics = lda\_model.fit\_transform(X\_count)

def display\_topics(model, feature\_names, no\_top\_words):

for topic\_idx, topic in enumerate(model.components\_):

print(f"Topic #{topic\_idx}:")

print(" ".join([feature\_names[i] for i in topic.argsort()[:-no\_top\_words - 1:-1]]))

print()

display\_topics(lda\_model, count\_vectorizer.get\_feature\_names\_out(), 10)

Conclusion:

The project successfully applied sentiment analysis and topic modeling techniques to Amazon product reviews. The models, including Multinomial Naive Bayes, SVM, and LSTM, were evaluated for their accuracy and effectiveness. Addressing class imbalance and optimizing models significantly improved prediction performance. Topic modeling using LDA provided insights into the underlying themes within the reviews, enhancing the overall analysis of consumer sentiments and topics.