SENTIMENT ANALYSIS ON CUSTOMER REVIEWS USING NLP FOR AMAZON REVIEW

OVERVIEW:

This project applies Natural Language Processing (NLP) to analyze customer reviews from the **Amazon Reviews Dataset for Musical Instruments**. The goal is to predict sentiment (positive, negative, or neutral) from text data, helping businesses improve product development, customer service, and marketing strategies.

KEY FEATURES:

- **Text Preprocessing:**

  Tokenization, stopword removal, stemming/lemmatization.

- **Feature Engineering:**

  TF-IDF, Bag-of-Words, Word2Vec.

- **Sentiment Classification:**

  Models trained to classify reviews as positive, neutral, or negative.

- **Modeling Techniques:**
  - Machine Learning: Logistic Regression, Random Forest, SVM.
  - Deep Learning: LSTMs, GRUs, and Transformer models (e.g., BERT).

- **Model Evaluation:**

  Accuracy, Precision, Recall, F1-Score, and AUC-ROC.

DATASET PREPARATION:

1. **Source**: The Amazon Customer Reviews Dataset (Musical Instruments category).

   Link: https://www.kaggle.com/datasets/eswarchandt/amazon-music-reviews/data?select=Musical_instruments_reviews.csv

WORKFLOW:

The sentiment analysis process for Amazon Musical Instruments reviews involves multiple stages, from data collection to model evaluation. Below is a step-by-step breakdown of the workflow:

1. Data Loading and Exploration

- **Dataset Selection**: The dataset focuses on customer reviews for musical instruments from the Amazon Reviews dataset.
- **Loading Data**: Use Python libraries (pandas, numpy) to load and inspect the dataset.
- **Exploration**: Perform exploratory data analysis (EDA) to understand:
    - Distribution of sentiment labels.
    - Common words or phrases in positive, neutral, and negative reviews.
    - Characteristics like review length, word count, and outliers.

2. Data Preprocessing

Text data requires cleaning and preparation to be used in machine learning models. Key steps include:

a. Text Cleaning

- Remove punctuation, special characters, and numerical values.
- Convert all text to lowercase for consistency.
- Remove unnecessary whitespaces.

b. Stopword Removal

- Use libraries like nltk or spaCy to remove frequently occurring words (e.g., "the", "and") that do not contribute to sentiment analysis.

c. Normalization

- Apply stemming or lemmatization to convert words to their base forms (e.g., "playing" → "play").

d. Handling Missing or Duplicate Data

- Remove rows with empty reviews or duplicate entries.

e. Label Encoding

- Map star ratings to sentiment classes:
    - **1-2 Stars**: Negative
    - **3 Stars**: Neutral
    - **4-5 Stars**: Positive

f. Class Balancing

- Use techniques like oversampling (e.g., SMOTE) or undersampling to address imbalances in sentiment distribution.

3. Feature Extraction

Convert the textual data into numerical representations using various techniques:

a. Bag-of-Words (BoW)

- Represent text as a sparse matrix of word frequencies or occurrences.

b. TF-IDF

- Use the **Term Frequency-Inverse Document Frequency** method to weigh terms based on their importance in the dataset.

c. Word Embeddings

- Apply pre-trained embeddings like **Word2Vec**, **GloVe**, or embeddings from **BERT** to capture semantic meaning and relationships between words.

4. Model Training

Train models using the preprocessed dataset and extracted features.

a. Baseline Models

- Train traditional machine learning models like:
    - **Logistic Regression**
    - **Random Forest**
    - **Support Vector Machine (SVM)**

b. Deep Learning Models

- Implement advanced models that capture sequential and contextual patterns in text:

- **Recurrent Neural Networks (RNNs)**
- **Long Short-Term Memory (LSTM)**
- **Gated Recurrent Units (GRUs)**

c. Transformer Models

- Fine-tune pre-trained models like **BERT** for better contextual understanding and state-of-the-art performance.

5. Model Evaluation

Assess model performance using metrics:

a. Classification Metrics

- **Accuracy**: Overall correctness.
- **Precision and Recall**: Measure the relevance and completeness of predictions.
- **F1-Score**: Balance between precision and recall, especially for imbalanced datasets.

b. AUC-ROC

- Evaluate the model's ability to distinguish between sentiment classes.

c. Confusion Matrix

- Analyze misclassification patterns and identify areas for improvement.

6. Prediction and Analysis

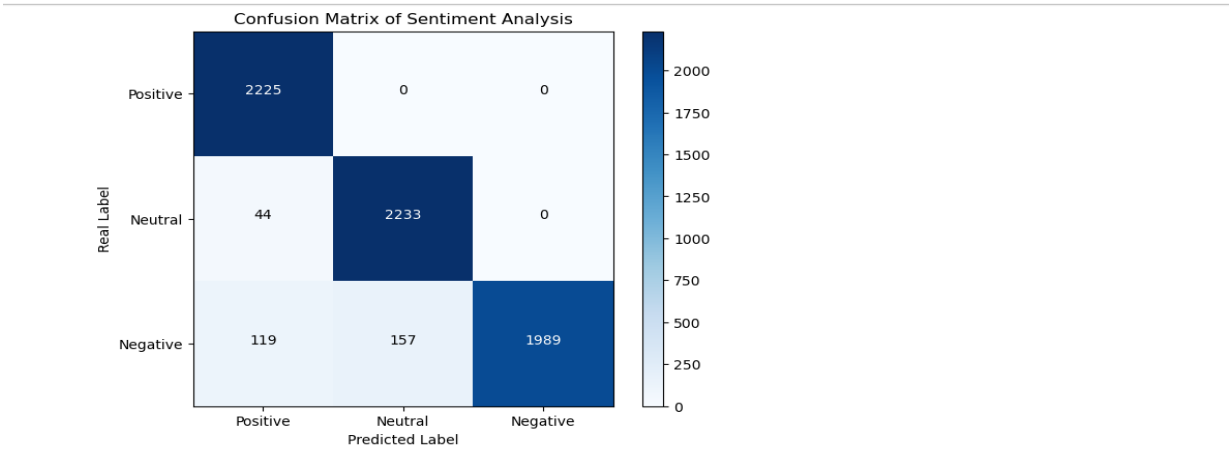
- Use the trained model to predict sentiment for unseen reviews.
- Visualize results using:
  - **Word Clouds**: Highlight frequently mentioned terms in positive or negative reviews.
  - **Bar Graphs or Pie Charts**: Display sentiment distribution across reviews.
  - **Trend Analysis**: Monitor changes in sentiment over time or across products.

7. Results and Insights

- Summarize the model's performance on test data.

- Provide actionable insights, such as:
    - Positive feedback themes for marketing.
    - Negative feedback themes for product improvement.
- Showcase visualizations to communicate findings effectively.

OUTPUT:



CLASSIFICATION REPORT

```
[64]: print(classification_report(y_test,Prediction))
```

	precision	recall	f1-score	support
0	0.93	1.00	0.96	2225
1	0.93	0.98	0.96	2277
2	1.00	0.88	0.94	2265
accuracy			0.95	6767
macro avg	0.96	0.95	0.95	6767
weighted avg	0.96	0.95	0.95	6767

CONCLUSION:

The **Sentiment Analysis on Amazon Reviews for Musical Instruments** project demonstrates the effective application of Natural Language Processing (NLP) techniques to derive actionable insights from textual customer feedback. By leveraging advanced models like transformers and employing robust preprocessing and feature extraction methods, this project provides a comprehensive approach to sentiment analysis.