Amazon_Unlocked_Mobile"

Problem Statement:

our goal is to perform sentiment analysis on customer reviews of client’s products on Amazon. The objective is to build a predictive model that can classify reviews into positive, negative, or neutral sentiments based on the expressed opinions. The analysis aims to understand the overall sentiment of customers towards the products and provide insights for product improvement, marketing strategies, and customer satisfaction."

Data Collection: data collected through web scraping, using Amazon’s APIs or services

Data Exploration and Understanding:

1.Loading The dataset: loading the dataset into Pandas DataFrame. This involves reading the dataset file in using amzon’s ApI’s. overview of the dataset size, the number of rows, and columns.

2.Basic Information: overview of the dataset, including column names, data types, and non-null counts.
**Use df.info() (Dataset Overview)65

Features: prodct name, brand name, price, rating, reviews, reviews votes

Missing Values:Checked for missing values in each column. Identifed columns with a significant number of missing values, especially in crucial columns such as ratings and reviews.
** df.isna().sum()
numeric columns null value filed with median and categorical column null values filed with mode and some null values dropped on the suggestion of client.

Text Quality:

Examine the quality of the text in reviews. Presence of spelling errors, abbreviations, inconsistent language can affect natural language processing tasks.

***text Preprocessing:

Lowercasing: Convert all text to lowercase to ensure consistency in the data and avoid treating words with different cases as different entities.

Tokenization:
Break down the text into individual words is called tokens. splits sentences into words, punctuation marks, and other meaningful words.

Removing Punctuation:punctuation marks like commas, periods, exclamation marks, and question marks to sentiment analysis and can be safely removed.

Removing Stopwords:Remove common stopwords, such as "and", "the", "is", "are", etc.reduces noise in the data.

Removing Numeric Characters:numerical characters In sentiment analysis, numerical values are not relevant to determining sentiment.

Stemming:Reduce words to their root form to normalize variations of the same word. remove prefixes or suffixes
Lemmatization:maps words to their base or dictionary form.

*Handling Abbreviations and Acronyms to their full forms for better understanding
*Correct spelling errors to improve the quality of the text helps in reducing noise .
* Removing HTML Tags strip them off to ensure clean text
*Convert emojis and special characters into textual representations or remove them altogether, depending on their relevance to the analysis.

**- Identify and handle rare words or typos appropriately. You may choose to remove them if they occur infrequently and do not contribute meaningfully to the analysis.

* After preprocessing, concatenate and join the tokens back into coherent text for further analysis.

Data Labeling: the process of assigning categories or labels to instances in a dataset,labeling for sentiment positive, negative, or neutral to customer reviews

Split the Dataset:
Divided the dataset into two subsets: one for training and the other for testing. The typical split ratio is 70-30 or 80-20, useing train _test_split function from scikitlearn
x: Reviews , y: Sentiment

****Feature Extraction***

Tokenization: Split each review into individual words(tokens).
Counting: Count the occurrence of each word in the review(document).
Vectorization: Represent each review as a vector where each element corresponds to the frequency of a word.

*Term Frequency-Inverse Document Frequency (TF-IDF)
TF-IDF calculates a word in a document relative to its frequency across all documents.

Term Frequency (TF): Measures the frequency of a word in a document.
Inverse Document Frequency (IDF): Measures the rarity of a word across all documents.
TF-IDF Score: Combines TF and IDF to assign a weight to each word in the document.

*N-grams:
N-grams are continuous sequences of N words in a document. They capture local context and syntactic information.

Unigrams (N=1): Single words as features.
Bigrams (N=2): Pairs of adjacent words.
Trigrams (N=3): Triplets of adjacent words.

**Part-of-Speech (POS) Tags: assigns grammatical categories (e.g., noun, prnoun,verb, adjective) to

words in a sentence.
tag each word in the text with its corresponding POS category.
Named Entity Recognition (NER)

**Word Embeddings:
Word embeddings represent words as dense, low-dimensional vectors in a continuous space. They capture semantic relationships between words.
Use pre-trained word embeddings (e.g., Word2Vec, GloVe) trained on large text corpora.
Train word embeddings specific to the dataset using techniques like Word2Vec or FastText.

****Model Selection:
i tested multiple models of Naive Bayes, Support Vector Machines (SVM), Logistic Regression,random forest,decision tree,KNN and checked...based on the accuracy score my team suggested three models to our client which has highest accuracy score.on top of that our client selected random forest based on project requirements

***Model Training:
Trained the selected model using the training dataset. During training, the model learns to map input features (text data) to sentiment labels (positive, negative, neutral).

Evaluation Metrics:
Evaluated the trained model’s performance on the testing set using metrics like accuracy, precision, recall, and F1-score.
Analyzed the confusion matrix to understand the model’s performance on different sentiment classes.

Model Interpretation:

Interpret the trained model to understand which features contribute to its predictions. Visualize important features or word clouds to gain insights into sentiment analysis results.

7. Cross-Validation (Optional):

Performed cross-validation to validate the model’s robustness and generalization ability. This involves splitting the dataset into multiple subsets and training the model on different combinations of training and validation sets.

8. Hyperparameter Tuning (Optional):

Fine-tuned the model’s hyperparameters further using techniques like grid search or random search to improve performance.

9. Model Deployment (Optional):
Deploy the trained model for making real-time predictions on new Amazon reviews. This may involve integrating the model into a web application or API for user interaction.

10. Continuous Improvement:
Monitor the model’s performance over time and retrain it with new data if necessary. Consider using more advanced techniques or larger datasets for better performance.
The model training process is iterative and may require experimentation with different algorithms, hyperparameters, and preprocessing techniques to achieve the desired performance. Continuously evaluate and refine the model based on feedback and new insights gained from the data.