Sentiment Analysis Using Naive Bayes

Overview

The project demonstrates a sentiment analysis model applied to restaurant reviews using the Naive Bayes classifier. The goal is to predict whether a review is positive or negative based on textual features extracted from the reviews.

Dataset

The dataset used for this analysis is the `Restaurant_Reviews.tsv`, which contains restaurant reviews and their corresponding sentiment labels. The dataset has the following structure:

- **Review**: The text of the review.
- **Liked**: Binary indicator where `1` represents a positive sentiment and `0` represents a negative sentiment.

Data Preprocessing and NLP

1. Text Cleaning:

- **Regular Expression (Regex) Filtering**: Non-alphabetic characters are removed from the text to ensure only words are processed.
- **Lowercasing**: Converts all characters in the text to lowercase to maintain uniformity.
- **Tokenization**: The cleaned text is split into individual words (tokens) for further processing.

2. Stopwords Removal:

- **Stopwords**: Common words that do not contribute much to the meaning of a sentence (e.g., 'the', 'is') are removed.
- **Exclusion of 'not'**: The word 'not' is retained to preserve negations which are crucial for accurate sentiment analysis (e.g., "not good").

3. Stemming:

- **Porter Stemmer**: Words are reduced to their root forms (e.g., 'running' becomes 'run') using the Porter Stemmer. This helps in normalizing different forms of a word to a common base.

Feature Extraction

The `CountVectorizer` from scikit-learn is used to convert the processed text into numerical features:

- **Bag of Words Mode**l: Represents each review as a vector of word counts. This model creates a matrix where each row corresponds to a review, and each column represents a word from the entire corpus. The value in the matrix represents the count of that word in the review.

Model Building

A Naive Bayes Classifier (GaussianNB) is employed to classify the reviews:

- **Training**: The model is trained on the training set, learning the relationship between the feature vectors and the sentiment labels.
- **Prediction**: The model predicts the sentiment of the reviews in the test set.

Evaluation

The performance of the model is evaluated using:

- **Confusion Matrix**: Shows the true positives, false positives, true negatives, and false negatives.
- **Accuracy Score**: Measures the proportion of correctly classified reviews.

Accuracy: 67.3%

Future Work

1. Model Improvement:

- Experiment with other classifiers (e.g., Support Vector Machines, Random Forests).
- Explore advanced text processing techniques like TF-IDF or Word Embeddings.

2. Hyperparameter Tuning:

- Optimize the model's hyperparameters to improve accuracy.

Conclusion

This project provides a foundational approach to sentiment analysis using textual data and machine learning. The Naive Bayes classifier, while simple, offers a good starting point for further experimentation and improvement in sentiment prediction tasks. The NLP preprocessing steps ensure that the text data is cleaned and transformed effectively to be used by the machine learning model.