**Text-Mining-Strategy for Sentiment-Classification in Product-Reviews**

**Introduction**

This research describes a sophisticated text-mining approach for sorting product evaluations into positive, negative, and neutral categories. This strategy allows retailers to acquire useful insights from client feedback by utilizing modern natural language processing (NLP) tools. Understanding consumer sentiment through reviews is critical for improving goods and services, increasing customer happiness, and accelerating corporate growth.

The technique consists of several critical steps: data gathering, preprocessing, sentiment analysis, and visualization. Initially, product reviews are gathered from Kaggle. These reviews are then preprocessed to clean and normalize the text data, making it suitable for analysis.

The core of the strategy is sentiment analysis, where machine learning algorithms and NLP models classify the reviews into positive and negative categories. These models are trained on labeled datasets and fine-tuned to accurately capture the nuances of customer sentiments.

This text-mining approach not only helps in understanding customer emotions but also in making data-driven decisions to enhance product offerings and customer experience.

**Data Loading and Overview**

In our text-mining project, we have successfully loaded a dataset into MongoDB, which serves as our database for storing and analyzing product reviews. The details of the database and the dataset are as follows:

**Database and Collection Details**

* **Database Name**: Bl\_Project
* **Collection Name**: BI\_Product\_Review

A screenshot of a computer

Description automatically generated

Creating New Instance

A screenshot of a computer

Description automatically generated

Creating Database

**Dataset Description**

The imported dataset contains reviews for 12 different products, with almost 20 reviews per product, making a total of 239 records. The dataset was randomly selected from the "Flipkart Product reviews with sentiment" dataset available on Kaggle. Each record in the dataset consists of the following columns:

1. **Product Name**: The name of the product being reviewed.
2. **Product Price**: The price of the product.
3. **Product Rating**: A numerical rating of the product, ranging from 1 to 5.
4. **Product Review**: A brief review of the product.
5. **Product Detailed Review**: A more detailed and comprehensive review of the product.
6. **Sentiment**: The sentiment associated with the review, categorized as positive or negative.

(vaghani, 2023)

**Data Loading Process**

The data loading process involved several steps to ensure that the dataset was correctly imported into MongoDB:

1. **Data Preparation**: The dataset was prepared by ensuring it was in a format suitable for MongoDB import. This included cleaning the data and converting it into CSV format.
2. **Database and Collection Creation**: A new database named Bl\_Project was created in MongoDB. Within this database, a collection named BI\_Product\_Review was established to store the product reviews.
3. **Data Import**: The prepared dataset was imported into the BI\_Product\_Review collection using MongoDB's import capabilities. This step involved using MongoDB commands or tools like mongoimport to load the CSV data into the collection.

A screenshot of a computer

Description automatically generated

Importing Data

A screenshot of a computer

Description automatically generated

Data in MongoDB

**Data Access and Preparation**

To examine the product reviews, we must first access the information contained in our MongoDB database.

We utilize Python's pymongo module to connect to the MongoDB database. By connecting to the local MongoDB instance, we can access the appropriate database and collection. This connection allows us to obtain the data contained in the BI\_Product\_Review collection.

After connecting to MongoDB, we import data from the BI\_Product\_Review collection into a Pandas DataFrame.

**Data Preprocessing**

By performing these preprocessing steps, we ensure that the textual data is in a suitable format for further analysis, enhancing the quality and reliability of the insights derived from the reviews.

Checking for Null Values

As a first step in data preparation, we looked for null values in our dataset. To preserve data integrity and accuracy in our analysis, we must ensure that there are no missing values. Fortunately, no null values were discovered in our data, allowing us to proceed without additional imputation or data cleaning in this respect.

Text Preprocessing

To prepare the textual data for analysis, we implemented a comprehensive text preprocessing function that performs several key operations:

1. **HTML Tag Removal**:
   * The function removes any HTML tags present in the reviews, ensuring that only the textual content is retained.
2. **Non-Alphabetic Character Removal**:
   * All characters except alphabetic ones are removed. This step eliminates numbers, punctuation, and special characters, which are generally not useful for sentiment analysis.
3. **Case Normalization**:
   * The text is converted to lowercase. This normalization step ensures that words like "Great" and "great" are treated as the same word, reducing redundancy.
4. **Stopword Removal**:
   * Common stopwords (e.g., "is", "are", "the") are filtered out using a predefined set. Removing these words helps focus on the more meaningful content of the reviews.
5. **Stemming**:
   * Stemming is applied to reduce words to their root forms. This process helps in normalizing variations of words (e.g., "running", "ran", "runner" are all reduced to "run"), which aids in consistent text analysis.
6. **Reconstruction**:
   * The cleaned, lowercased, and stemmed words are then joined back into a single string for each review, making the text ready for further analysis.

The preprocessing function was applied to our dataset's 'product detailed review' column. The systematic conversion of raw reviews into cleaned and normalized text is critical for reliable sentiment analysis and other natural language processing applications.

The distribution of reviews in to Positive and Negative:

A blue and orange bar graph

Description automatically generated

Distribution of Sentiment

**Text Vectorization and Model Training**

TF-IDF Vectorization

We utilized a TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer to transform the textual data from product\_detailed\_review into numerical representations. This vectorizer calculates the importance of each word in relation to its occurrence across all reviews, thereby converting text data into a sparse matrix format. The resulting reviews\_corpus variable represents reviews as rows and unique words as columns, weighted by their TF-IDF scores. This preprocessing step prepares the data for subsequent machine learning model training.

Train-Test Split

We partitioned the TF-IDF transformed data (reviews\_corpus) and the corresponding sentiment labels into training (X\_train, Y\_train) and testing (X\_test, Y\_test) sets using train\_test\_split from scikit-learn. We allocated 30% of the data for testing to evaluate the models' performance, ensuring a fixed random seed (random\_state=42) for reproducibility.

Model Training and Evaluation

We trained two classification models on the training data:

1. Multinomial Naive Bayes Classifier
2. Random Forest Classifier

Both models were trained to predict sentiment labels based on the TF-IDF transformed reviews. After training, the models were evaluated on the testing set to assess their accuracy.

Results

Both the Multinomial Naive Bayes and Random Forest classifiers achieved an accuracy of 87.50% on the testing data, highlighting their effectiveness in sentiment classification. The consistency in confusion matrices further validates their performance across sentiment categories.

A close up of black text

Description automatically generated

Confusion matrix

**Conclusion**

In conclusion, this report presents a comprehensive text-mining approach aimed at analyzing product reviews for sentiment classification into positive and negative categories. Leveraging MongoDB for data storage and Python for data access and preprocessing, we successfully transformed raw textual data into structured insights ready for analysis. Through TF-IDF vectorization and training of Multinomial Naive Bayes and Random Forest classifiers, we achieved an impressive 87.50% accuracy in sentiment prediction on our testing dataset. This methodology not only enhances our understanding of customer sentiment but also equips businesses with actionable insights to improve products, enhance customer satisfaction, and drive strategic decision-making. By integrating modern NLP techniques with robust machine learning models, this approach demonstrates its potential to support data-driven strategies in enhancing overall business performance and customer experience.

References

vaghani, N. (2023, February 3). *Flipkart product reviews with sentiment dataset*. Kaggle. https://www.kaggle.com/datasets/niraliivaghani/flipkart-product-customer-reviews-dataset