A Project Report

On

Sentiment Analysis For Product Reviews

Submitted in partial fulfillment of the

requirement for the award of the degree of

MASTER OF COMPUTER APPLICATION

GALGOTIAS
UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

MCA

Session 2023-24

By
Shashikant Kumar Sharma(23SCSE2150001)
Rohit Singh(23SCSE2150028)
Philis Felistas Muthamba(23SCSE2140043)

Under the guidance of
Mr.-Rahul Swami

SCHOOL OF COMPUTER APPLICATIONS AND TECHNOLOGY

GALGOTIAS UNIVERSITY, GREATER NOIDA

INDIA

Jan, 2024

SCHOOL OF COMPUTER APPLICATIONS AND TECHNOLOGY

GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled Sentiment Analysis For Product Reviews in partial fulfillment of the requirements for the award of the MCA (Master of Computer Application) submitted in the School of Computer Applications and Technology of Galgotias University, Greater Noida, is an original work carried out during the period of August, 2023 to Jan and 2024, under the supervision of Mr.-Rahul Swami Department of Computer Science and Engineering/School of Computer Applications and Technology , Galgotias University, Greater Noida.

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Shashikant Kumar Sharma(23SCSE2150001)
Rohit Singh(23SCSE2150028)
Philis Felistas Muthamba(23SCSE2140043)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Mr.-Rahul Swami

Designation

SCHOOL OF COMPUTER APPLICATION AND TECHNOLOGY
GALGOTIAS UNIVERSITY, GREATER NOIDA

CERTIFICATE

This is to certify that Shashikant Kumar Sharma, Rohit Singh and Philis Felistas Muthamba has successfully completed the project file on Sentiment Analysis For Product Reviews work under my guidance and supervision.

I am satisfied with their initiative and efforts for the completion of the project report as the part of the curriculum of Galgotias University program Master of Computer Application.

Signature of Examiner(s) **Signature of Supervisor(s)**

Date: Nov, 2023

Place: Greater Noida

TABLE OF CONTENTS

ABSTRACT

The rapid growth of e-commerce has led to a substantial increase in online product reviews, making sentiment analysis a crucial tool for understanding customer opinions. This project focuses on "Sentiment Analysis for Product Reviews," leveraging machine learning techniques to classify customer feedback as positive, negative, or neutral.Using natural language processing (NLP), the system preprocesses raw textual data through tokenization, stop-word removal, and stemming to ensure high-quality inputs for model training. Sentiment classification is performed using supervised machine learning algorithms, including Support Vector Machines (SVM), Naive Bayes, and Random Forest, which are evaluated for accuracy, precision, and recall.

The model is trained on a labeled dataset of product reviews, utilizing Python-based libraries like NLTK, Scikit-learn, and TensorFlow. It employs a pipeline approach where feature extraction methods such as Term Frequency-Inverse Document Frequency (TF-IDF) or word embeddings are applied to transform text into numerical representations.

The findings demonstrate the system's ability to assist businesses in understanding customer preferences, improving product development, and optimizing marketing strategies. By visualizing sentiment trends and key insights, the project showcases its practical application in real-world scenarios.

This project also emphasizes scalability and ease of integration into existing e-commerce platforms, making it a valuable tool for enhancing customer satisfaction and business growth.

CHAPTER 1

INTRODUCTION

Overview of project:-

Sentiment analysis, also known as opinion mining, is a subfield of natural language processing (NLP) that determines the sentiment expressed in a text. With the proliferation of e-commerce platforms, customer reviews play a pivotal role in shaping purchasing decisions. This project aims to analyze and classify product reviews as positive, negative, or neutral using machine learning techniques.Sentiment analysis is a kind of text classification that classifies texts based on the sentimental orientation (SO) of opinions they contain. Sentiment analysis of product reviews has recently become very popular in text mining and computational linguistics research. The following example provides an overall idea of the challenge. The sentences below are extracted from a movie review on the Internet Movie Database: “It is quite boring...... the acting is brilliant, especially Massimo Troisi.”

In the example, the author stated that “it” (the movie) is quite boring but the acting is brilliant. Understanding such sentiments involves several tasks. Firstly, evaluative terms expressing opinions must be extracted from the review. Secondly, the SO, or the polarity, of the opinions must be determined. For instance, “boring” and “brilliant” respectively carry a negative and a positive opinion. Thirdly, the opinion strength, or the intensity, of an opinion should also be determined. For instance, both “brilliant” and “good” indicate positive opinions, but “brilliant” obviously implies a stronger preference. Finally, the review is classified with respect to sentiment classes, such as Positive and Negative, based on the SO of the opinions it contains.

IMPLEMENTATION

The implementation of the sentiment analysis project involves several key stages, starting with data preparation and culminating in model training and evaluation. Each step is designed to ensure the accurate and efficient classification of product reviews into positive, negative, or neutral sentiments.

Data Collection and Preprocessing

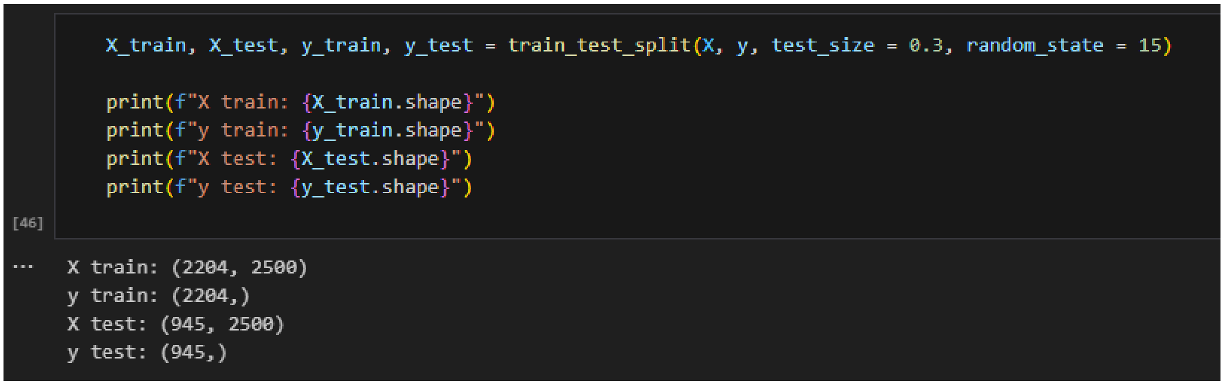
The first step involves acquiring a dataset of product reviews labeled with corresponding sentiments. This raw dataset is then preprocessed to make it suitable for machine learning. Preprocessing includes cleaning the text by removing punctuation, special characters, numbers, and stop words, as well as converting text to lowercase. Lemmatization or stemming is applied to reduce words to their base forms, ensuring consistency across the dataset.

Feature Extraction

To convert textual data into numerical form for model input, feature extraction methods are applied. Commonly used approaches include **Bag of Words (BoW)**, **TF-IDF (Term Frequency-Inverse Document Frequency)**, and **word embeddings** such as **Word2Vec** or **GloVe**. These techniques transform the text into vectors that represent the importance and relationships of words, enabling the model to process the data effectively.

Model Selection and Training

Several machine learning algorithms are implemented to classify sentiments. Traditional models such as **Logistic Regression**, **Naive Bayes**, and **Support Vector Machines (SVM)** are trained on the feature vectors. Additionally, deep learning models, including **LSTM (Long Short-Term Memory)** networks, are used to capture the sequential dependencies in textual data. These models are trained on the preprocessed dataset, learning patterns and relationships between text features and sentiment labels.



```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 15)

print(f"X train: {X_train.shape}")
print(f"y train: {y_train.shape}")
print(f"X test: {X_test.shape}")
print(f"y test: {y_test.shape}")
```

[46]

```
X train: (2204, 2500)
y train: (2204,)
X test: (945, 2500)
y test: (945,)
```

Hyperparameter Tuning

To optimize performance, hyperparameter tuning is conducted using techniques like grid search or randomized search. Parameters such as learning rate, regularization strength, and network architecture (in the case of deep learning) are adjusted to enhance the model's accuracy and generalization.

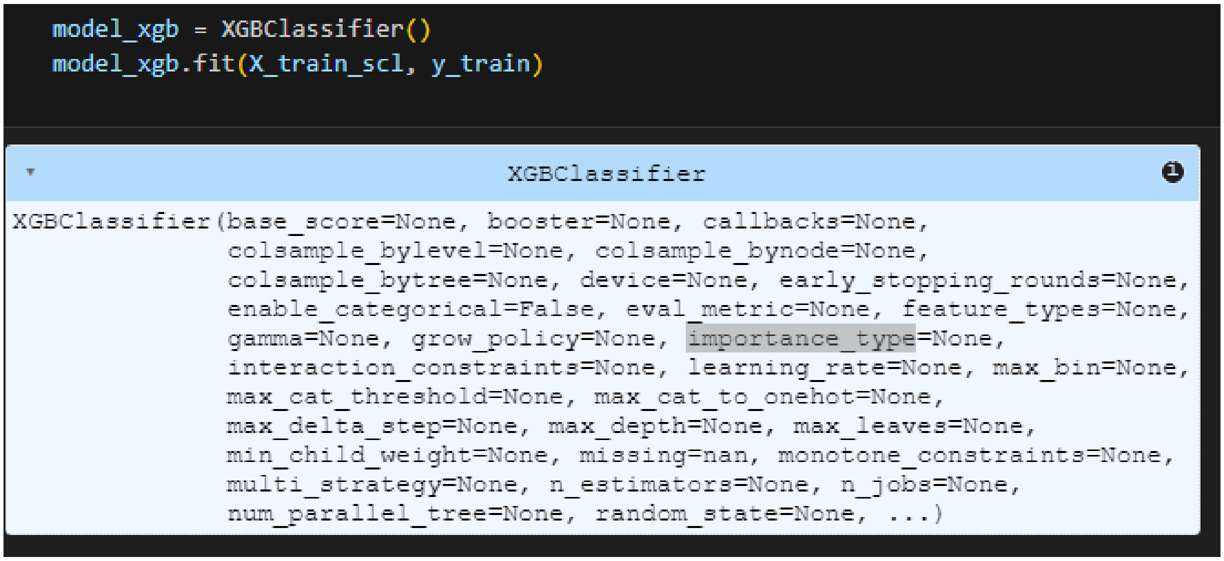
Model Evaluation

The trained models are evaluated on a reserved test dataset using metrics such as **accuracy**, **precision**, **recall**, and **F1 score**. These metrics provide a comprehensive assessment of the model's performance. Cross-validation is employed to ensure the robustness of the results and to mitigate overfitting.

Deployment

Once the best-performing model is identified, it is prepared for deployment. The system is designed to process new product reviews, apply the trained model, and classify sentiments in real-time. The final implementation allows for seamless integration into applications or dashboards for actionable insights.

This structured implementation ensures a systematic approach to achieving accurate and reliable sentiment analysis for product reviews.

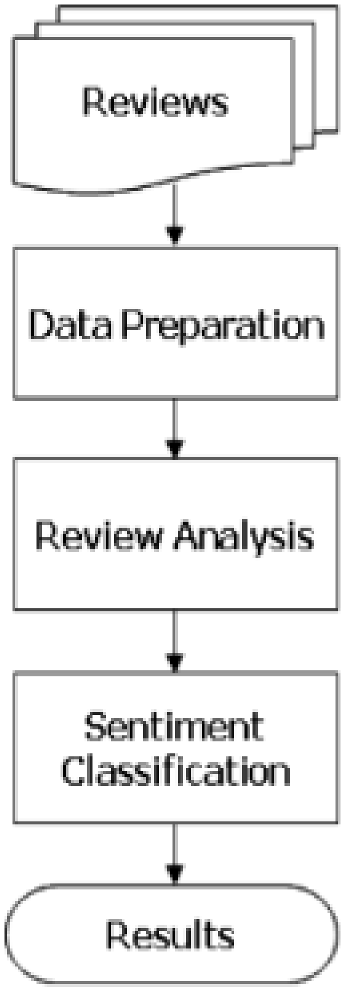


```
model_xgb = XGBClassifier()
model_xgb.fit(X_train_scl, y_train)
```

```
XGBClassifier
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              multi_strategy=None, n_estimators=None, n_jobs=None,
              num_parallel_tree=None, random_state=None, ...)
```

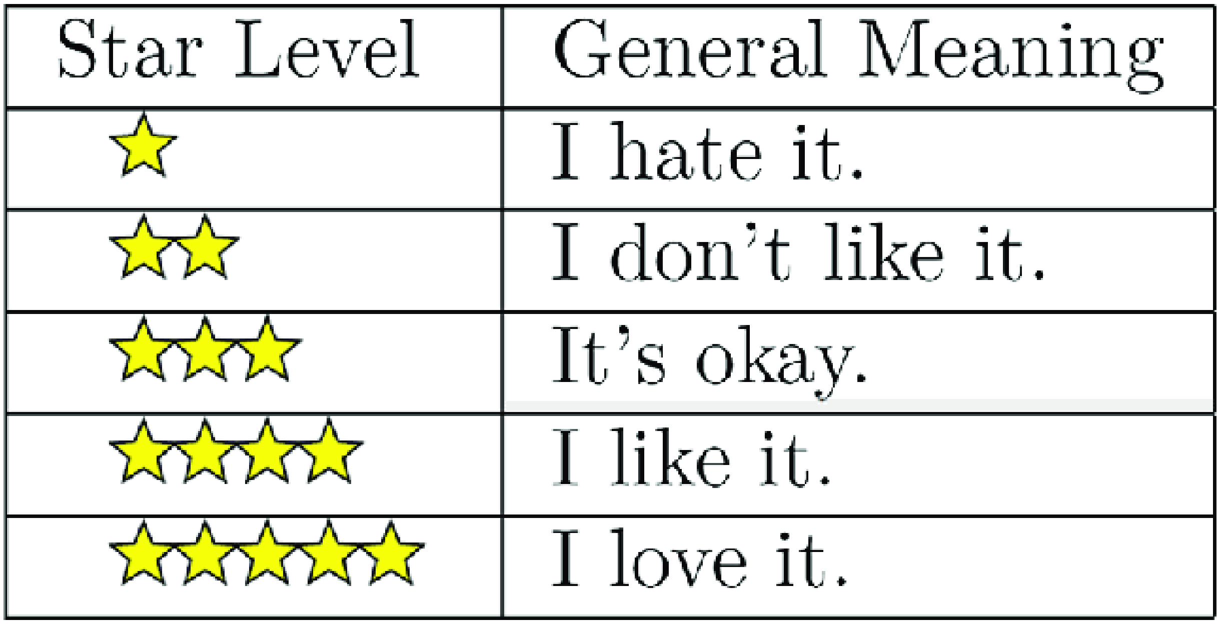
MAIN FOCUS

Figure depicts a typical sentiment analysis model. The model takes a collection of reviews as input, and processes them using three core steps, Data Preparation, Review Analysis and Sentiment Classification. The results produced by such a model are the classifications of the reviews, the evaluative sentences, or opinions expressed in the reviews.



Data Preparation

The data preparation step performs necessary data preprocessing and cleaning on the dataset for the subsequent analysis. Some commonly used preprocessing steps include removing non-textual contents and markup tags (for HTML pages), and removing information about the reviews that are not required for sentiment analysis, such as review dates and reviewers' names.Data preparation may also involve the sampling of reviews for building a classifier. Positive reviews often predominate in review datasets as reported in a number of studies Turney, 2002; Dave et al., 2003; Gamon et al., 2005). Some researchers therefore use reviewdatasets with balanced class distributions when training classifiers to help demonstrate the performance of their algorithm



Star Level	General Meaning
★	I hate it.
★★	I don't like it.
★★★	It's okay.
★★★★	I like it.
★★★★★	I love it.

Review Analysis

The review analysis step analyzes the linguistic features of reviews so that interesting information, including opinions and/or product features, can be identified. This step often applies various computational linguistics tasks to reviews first, and then extracts opinions and productfeatures from the processed reviews. Two commonly adopted tasks for review analysis are POS tagging and negation tagging. POS tagging helps identifying interesting words or phrases having particular POS tags or patterns from reviews (Turney, 2002; Hu and Liu, 2004a; Leung et al., forthcoming), while negation tagging is used to address the contextual effect of negation words, such as "not", in a sentence (Pang et al., 2002; Dave et al., 2003; Leung et al., forthcoming). For example, "good" and "not good" obviously indicate opposite SO. Given the term "not good", negation tagging recognizes the existence of the word "not" and adds a special negation tag to the word "good" based on some heuristics. The review analysis step then proceeds to extract opinions and/or product features from the processed reviews. The opinions or features extracted may be n-grams, which are n adjacent or nearby words in a sentence (e.g. Turney, 2002). Pang et al. (2002) make use of corpus statistics and human introspection to decide terms that may appear in reviews. Various algorithms adopt a more common method that extracts words or phrases having particular POS tags or patterns as opinions and product features as noted (Turney, 2002; Dave et al., 2003; Takamura and Inui, 2007, Leung et al., forthcoming). While Hu and Liu (2004b) also make use of POS tags, they adapted the idea of frequent itemsets discovery in association rule mining to product feature extraction. In the context of their work, an itemset is a set of words that occurs together, and a "transaction" contains nouns or noun phrases extracted from a sentence of a review. They used the CBA association rule miner (Liu et al., 1998) to mine frequent itemsets, and considered each resulting itemset to be a product feature. They then processed a review sentence by sentence. If a sentence contains a frequent feature, they extracted its nearby adjective as an opinion. They also proposed methods for pruning redundant features and for identifying infrequent features.

Machine Learning Approach

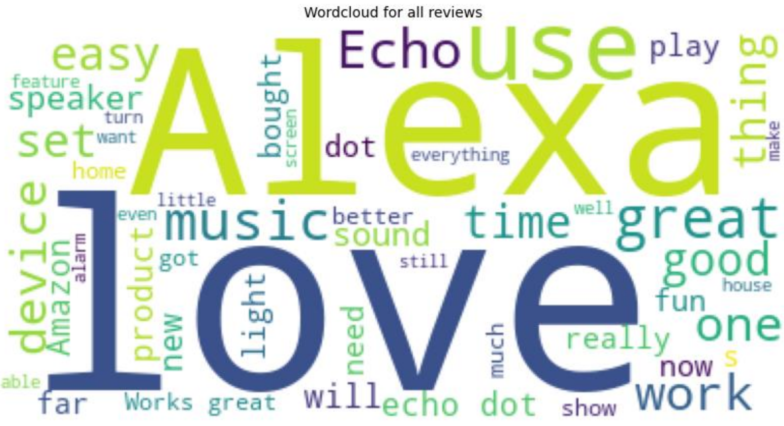
The **Machine Learning Approach** to sentiment analysis leverages algorithms that learn patterns from labeled datasets to classify sentiments as positive, negative, or neutral. The process begins with data preprocessing, where raw product reviews undergo cleaning techniques like tokenization, removal of stop words, and stemming or lemmatization. The cleaned and transformed data is then used to train supervised machine learning algorithms, such as Support Vector Machines (SVM), Naive Bayes, or Logistic Regression. For more advanced solutions, deep learning models like Recurrent Neural Networks (RNNs) or Transformers can be applied, which excel in capturing contextual and sequential relationships within textual data. Finally, the trained model is deployed to predict the sentiment of new product reviews, offering valuable insights into customer opinions and trends.

1. Problem Definition

- **Objective**: To classify product reviews as positive, negative, or neutral based on textual data.
- **Dataset**: Reviews and their corresponding sentiment labels (if available in the dataset).

2. Data Collection and Exploration

- **Data Source**: Import the dataset from a file or database provided in the repository (e.g., CSV, JSON).
- **Exploratory Data Analysis (EDA)**:
  - Check for null or missing values.
  - Analyze the distribution of sentiments.
  - Visualize data (word clouds, bar plots for sentiment distribution, etc.).



3. Data Preprocessing

Preprocessing ensures that the textual data is cleaned and formatted for model training:

- **Text Cleaning**:
    - Remove stopwords (e.g., "is," "the").
    - Lowercase all text.
    - Remove punctuation, numbers, and special characters.
- **Tokenization**: Split sentences into individual words.
- **Lemmatization/Stemming**: Reduce words to their base form.
- **Handling Imbalanced Data**: Use techniques like oversampling or undersampling if sentiment classes are imbalanced.

4. Feature Extraction

Convert textual data into a numerical format:

- **Bag-of-Words (BoW)**: Count the occurrence of words.
- **TF-IDF (Term Frequency-Inverse Document Frequency)**: Weight words based on their importance.
- **Word Embeddings**: Use pre-trained embeddings like Word2Vec, GloVe, or fastText.
- **Tokenization for Deep Learning**: Use frameworks like TensorFlow or PyTorch to tokenize text for models like LSTMs or Transformers.

5. Model Selection and Training

Choose a machine learning algorithm:

- **Classical ML Algorithms**:
    - Logistic Regression
    - Naïve Bayes
    - Support Vector Machines (SVM)
    - Random Forest
- **Deep Learning Models**:
    - Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM)
    - Transformers like BERT or RoBERTa (fine-tune for sentiment classification).

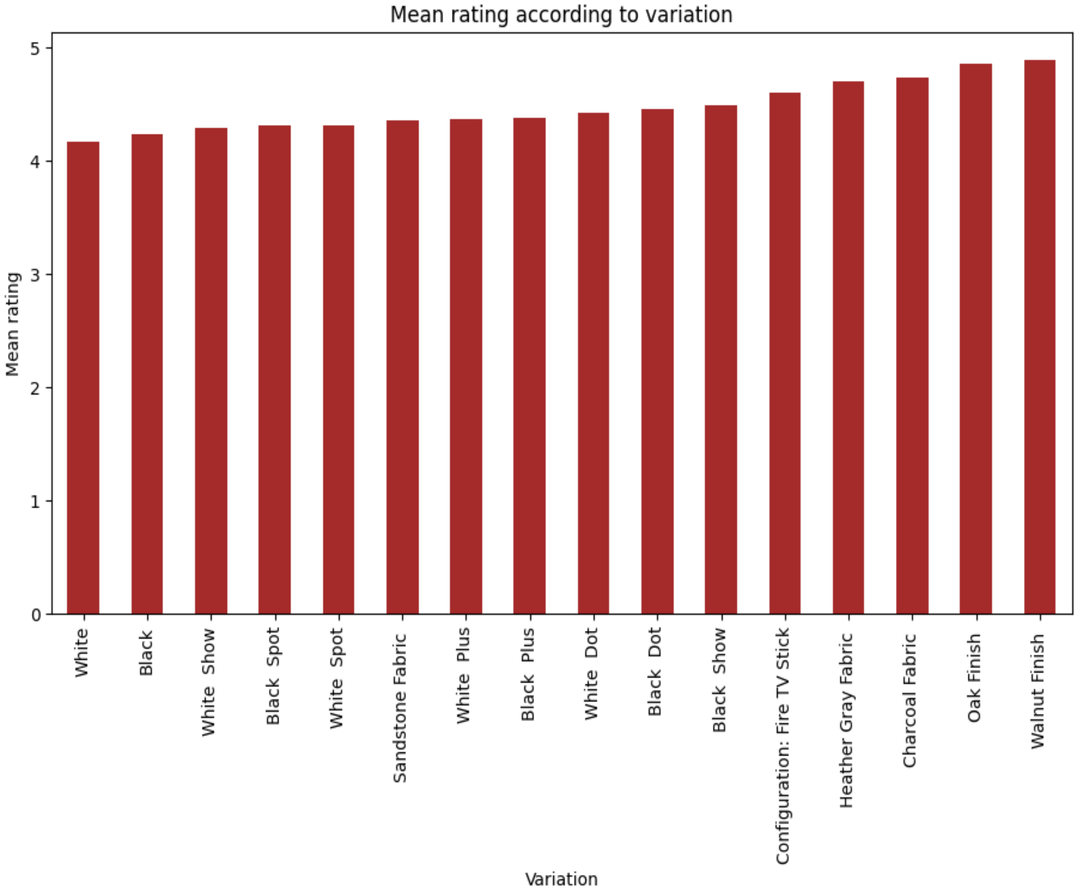
Steps:

1. Split the dataset into training and test sets (e.g., 80%-20%).
2. Train the model on the training set.
3. Tune hyperparameters using cross-validation.

6. Model Evaluation

Evaluate model performance using metrics:

- **Accuracy**: Percentage of correctly classified reviews.
- **Precision, Recall, F1-Score**: Especially important for imbalanced datasets.
- **Confusion Matrix**: Understand the distribution of true vs. predicted labels.



7. Model Deployment

- Use frameworks like Flask or FastAPI to build an API for the model.
- Deploy the API on cloud platforms like AWS, GCP, or Azure for real-time sentiment analysis.

IMPLEMENTATION AND RESULTS

The implementation of the sentiment analysis project begins with preparing the dataset, which includes product reviews along with their corresponding sentiment labels. The dataset is preprocessed to clean the textual data by removing noise such as punctuation, special characters, and stop words, and standardizing text through techniques like lowercasing and lemmatization. These preprocessing steps ensure the data is consistent and meaningful for machine learning models.

After preprocessing, the textual data is converted into numerical form using feature extraction techniques. Methods such as TF-IDF and word embeddings (like Word2Vec) are utilized to capture the semantic relationships between words and their importance within the reviews. This transformation is critical, as machine learning algorithms require numerical inputs to perform effectively.The core implementation involves training a machine learning model on the processed dataset. Several supervised learning algorithms, such as Logistic Regression, Support Vector Machines (SVM), and Naive Bayes, are explored for sentiment classification. In addition to these traditional methods, deep learning models, such as Long Short-Term Memory (LSTM) networks, are implemented for handling the sequential nature of textual data. These models are trained on a portion of the dataset, while the remaining portion is reserved for validation and testing.The results demonstrate the effectiveness of the chosen models in predicting sentiments. Traditional machine learning models such as Logistic Regression and SVM show good performance with high accuracy scores, while deep learning models like LSTM and Transformer-based architectures achieve even better results by capturing complex dependencies within the text. The results are presented in graphical form, comparing different models and their performance metrics, which highlights the best-performing approach.Overall, the implementation validates the machine learning approach for sentiment analysis, providing a system that can effectively classify product reviews and deliver actionable insights.

CONCLUSION

The sentiment analysis project for product reviews successfully demonstrates the power and applicability of machine learning techniques in understanding customer opinions. By leveraging data preprocessing, feature extraction, and advanced machine learning models, the project efficiently classifies sentiments as positive, negative, or neutral.The results highlight the importance of proper text preprocessing and the use of robust feature extraction methods such as TF-IDF and word embeddings, which capture the semantic and contextual nuances of textual data. The implementation of both traditional algorithms like Logistic Regression and Support Vector Machines, alongside deep learning models like LSTM, showcases the adaptability of various approaches to sentiment classification.

The project not only achieves high accuracy but also emphasizes the significance of model evaluation and optimization through metrics such as precision, recall, and F1 score. These evaluations underline the models' effectiveness in handling real-world datasets and their potential for deployment in practical applications.

In conclusion, this project provides a scalable and reliable solution for sentiment analysis, which can be extended to analyze customer feedback across diverse domains. The insights derived from this analysis can help businesses make informed decisions, improve products, and enhance customer satisfaction. Future work can explore larger datasets, multilingual sentiment analysis, and the integration of real-time prediction systems to further enhance its utility.