iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

ENHANCED SENTIMENT ANALYSIS OF CUSTOMER REVIEWS

Jammula Vijayalakshmi[1]
Jampu Abhinaya[2]
Jonnalagadda Subhashini[3]
B. Tech Student, Dept. of Computer Science and Engineering, R.V.R & J.C College of Engineering, Chowdavaram, Guntur, Andhra Pradesh, India

Mrs. Zareena Noorbasha[4]
Assistant Professor, Dept. of Computer Science and Engineering, R.V.R & J.C College of Engineering, Chowdavaram, Guntur, Andhra Pradesh, India

ABSTRACT

The growing reliance on customer feedback for improving products and services has made sentiment analysis a crucial task for businesses. This study focuses on enhancing sentiment analysis using advanced machine learning and deep learning models. The system combines Random Forest, XGBoost, LSTM, CatBoost, and BERT—leveraging Hugging Face's Transformers library. Random Forest and XGBoost offer strong ensemble learning capabilities, LSTM captures long-term dependencies in text, CatBoost handles categorical variables efficiently, and BERT introduces bidirectional context-aware representations of text. With BERT's powerful transformer-based embeddings, the model improves its ability to interpret nuanced and complex sentiments, including sarcasm and idioms. The integration of these diverse models aims to deliver a highly accurate, robust, and scalable sentiment analysis framework, enabling businesses to gain actionable insights and enhance customer satisfaction.

Keywords:
Sentiment Analysis, Customer Reviews, Machine Learning, Random Forest, XGBoost, LSTM, CatBoost, BERT, Hugging Face Transformers, Ensemble Learning, NLP

INTRODUCTION

Sentiment analysis has become a key tool for businesses to gain valuable insights from customer reviews. As customer feedback continues to play a crucial role in shaping product and service offerings, analyzing sentiments expressed in these reviews becomes increasingly important. Traditional methods of sentiment analysis, however, often fall short when dealing with the complexity of language and the subtleties of customer feedback. This project aims to address these limitations by implementing a combination of advanced machine learning algorithms.

The integration of Random Forest, XGBoost, LSTM, and CatBoost allows for more accurate and robust sentiment classification. Random Forest provides a strong baseline with its ability to manage large datasets, while XGBoost enhances predictive performance. LSTM captures contextual dependencies, and CatBoost addresses the challenges of handling categorical and unstructured data. Together, these models offer a comprehensive solution to improving sentiment analysis, enabling businesses to better understand and respond to customer needs.

OBJECTIVES

Develop an advanced sentiment analysis system using Random Forest, XGBoost, LSTM, CatBoost, and BERT. Enhance classification accuracy, contextual awareness, and domain adaptability. Integrate Hugging Face's BERT model to improve understanding of sarcasm, negation, and complex syntactic structures. Deliver sentiment classification results (positive, negative, neutral) for varied customer review datasets.

Train and evaluate models on diverse customer review datasets. Incorporate BERT with Hugging Face Transformers for embedding generation. Combine BERT embeddings with traditional ML classifiers (Random Forest, XGBoost, CatBoost). Utilize LSTM for sequential context extraction. Evaluate model performance using Accuracy, Precision, Recall, and F1-Score. Deploy the solution in a Flask-based application with upload, prediction, and user management capabilities.

IJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

As businesses increasingly rely on customer feedback for product and service improvement, the demand for accurate sentiment interpretation is critical. Reviews often involve subtle emotions, complex sentence structures, and sarcasm. By integrating BERT from Hugging Face, which provides powerful transformer-based embeddings, alongside traditional and deep learning models, the system enhances contextual understanding. This topic is chosen to explore the synergistic potential of these models in real-world sentiment analysis applications.

LITERATURE SURVEY

[1]. Alharbi NM, Alghamdi NS, Alkhammash EH, Al Amri JF, "Evaluation of sentiment analysis via word embedding and RNN variants for Amazon online reviews," 2021. This paper by Alharbi et al. compares various approaches using embeddings and RNN architectures, focusing on their ability to capture the sentiment present in reviews.

[2]. Xia H, Yang Y, Pan X, Zhang Z, An W, "Sentiment analysis for online reviews using conditional random fields and support vector machines," 2020. The paper presents an approach to sentiment analysis for online reviews using Conditional Random Fields (CRFs) and Support Vector Machines (SVM).

[3]. Tang F, Fu L, Yao B, Xu W, "Aspect based fine-grained sentiment analysis for online reviews," 2019. This paper focuses on fine-grained sentiment analysis, specifically analyzing aspects of online reviews to understand the nuanced sentiments expressed.

[4]. Huang M, Xie H, Rao Y, Liu Y, Poon LK, Wang FL, "Lexicon-based sentiment convolutional neural networks for online review analysis," 2020. The paper explores a lexicon-based approach integrated with Convolutional Neural Networks (CNN) for sentiment analysis of online reviews.

[5]. Ghiassi M, Lee S, "A transferable lexicon set for Twitter sentiment analysis using a supervised machine learning approach," 2018. This paper presents a domain-transferable lexicon set designed for sentiment analysis on Twitter data using a supervised machine learning approach.

SYSTEM ANALYSIS

Existing System

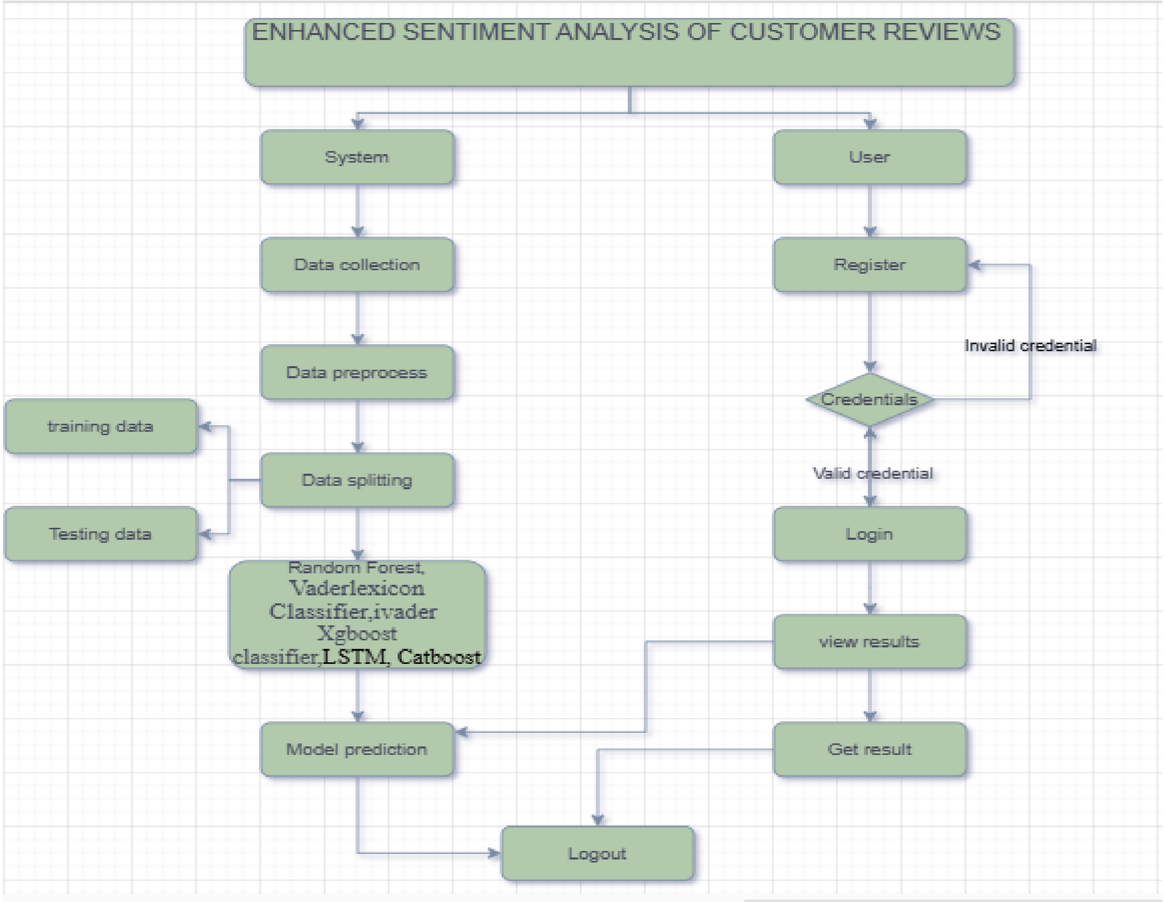
The existing system for sentiment analysis of customer reviews often relies on traditional lexicon-based approaches or basic machine learning models. One of the popular tools used in sentiment analysis tasks is the **VADER (Valence Aware Dictionary and sEntiment Reasoner)** lexicon classifier. VADER is a rule-based sentiment analysis model that uses a predefined dictionary of words associated with sentiment scores and combines this information with syntactical rules to classify text as positive, neutral, or negative.VADER is particularly well-suited for short texts like social media posts, tweets, or reviews, where emoticons, slang, and informal language are prevalent. It is designed to handle the complexities of real-world sentiment expressed in these domains, making it a useful tool for quick and relatively accurate sentiment classification tasks in various business applications, including customer reviews.However, despite its strengths, VADER has certain limitations that can impact its effectiveness in more complex or large-scale sentiment analysis tasks.

Disadvantages

- **Limited Contextual Understanding**: VADER uses predefined sentiment scores for words and relies on simple rules to classify sentiment. While this approach works well for short and straightforward texts, it lacks the ability to fully capture the contextual nuances of more complex sentences. For example, VADER may misinterpret sarcasm, irony, or negative sentiment expressed through positive words (e.g., "not bad" or "a great disaster").
- **Inability to Handle Long Texts**: VADER is optimized for short texts like tweets or product reviews. When applied to long-form content, it often fails to capture the sentiment accurately, especially if the sentiment changes throughout the text. In contrast, models like LSTM are capable of handling long-term dependencies and can better understand how sentiment evolves across sentences.
- **Inflexibility with Domain-Specific Language**: VADER is built on a generic lexicon, which may not perform well with domain-specific terminology or jargon. For example, in industries like healthcare, finance, or tech, specialized vocabulary or context might lead to incorrect sentiment classifications. Unlike machine learning models like XGBoost or CatBoost, VADER lacks the ability to adapt and learn from domain-specific data,

PROJECT FLOW

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/



Proposed system:
The enhanced system combines:
Random Forest: Strong baseline and interpretability
XGBoost: Excellent performance on structured data
LSTM: Captures temporal sequences in reviews
CatBoost: Handles categorical features and overfitting
BERT and Hugging Face Transformers: Captures bidirectional contextual embeddings and nuances in language using pre-trained transformer models like bert-base-uncased.

Advantages
Context-Aware Analysis via BERT
Strong Generalization from Ensemble Learning
Better Interpretability and Robustness
Domain Adaptability
Highly Accurate on Long & Sarcastic Texts

REQUIREMENT ANALYSIS

Functional Requirements
Data Collection and Preprocessing:
- Ability to collect and preprocess customer reviews data from multiple sources.
- Cleaning of data to remove noise, handle missing values, and normalize text (e.g., tokenization, stemming, lemmatization).

Sentiment Classification Models:
- Implementation of multiple machine learning models, including:

IJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

Random Forest: Used for feature selection and as a baseline model.
XGBoost: Employed to improve predictive accuracy.
LSTM: Used for capturing contextual dependencies in the text.
CatBoost: Optimized for categorical and unstructured data.

- Ability to train and evaluate these models on the customer review data.

· **Model Integration**:

- An ensemble approach that combines multiple models (e.g., Voting Classifier or Stacking Classifier) to improve the overall sentiment classification accuracy.
- The ability to choose the optimal ensemble method based on performance metrics.

· **Sentiment Analysis Output**:

- The models should output sentiment classifications (positive, negative, neutral) for each review.
- The ability to provide a confidence score for each sentiment classification.
- Ability to handle multi-class sentiment analysis if necessary.

· **Model Performance Evaluation**:

- Evaluate model performance using standard metrics like accuracy, precision, recall, F1-score, and confusion matrix.
- Ability to compare model performances and select the best-performing model.

· **User Interface (UI)**:

- A simple UI for inputting customer reviews and obtaining sentiment analysis results.
- Visualization of sentiment distribution for a batch of reviews (e.g., bar chart or pie chart).

Data Export/Integration:

- Ability to export the sentiment analysis results into different formats (e.g., CSV, JSON).
- Integration with existing business systems (e.g., CRM tools, feedback platforms) to automate sentiment analysis workflows.

Non-functional requirements

1. **Performance and Scalability**:

The system must be able to handle a large volume of customer reviews in a timely manner.
The solution must scale horizontally to accommodate increases in data size or user load.

2. **Accuracy**:

The sentiment analysis models should provide high accuracy, ideally with an error rate of less than 5% for basic sentiment classification tasks.
The models should be robust enough to handle diverse and noisy text data.

3. **Latency**:

Sentiment analysis should be performed in real-time or with minimal delay, allowing businesses to quickly respond to customer feedback.

4. **Robustness**:

The system must be resilient to noise in the data, such as slang, misspellings, or incomplete reviews. The models should handle edge cases and various sentiment expressions effectively.

5. **Security**:

Customer review data must be protected from unauthorized access and leakage.
Implement secure data storage and communication protocols.

6. **Maintainability**:

The system should be modular to allow easy updates and maintenance, including model retraining and the addition of new models or features. Codebase should be well-documented to facilitate future enhancements and troubleshooting.

7. **Usability**:

The system should be user-friendly, with minimal technical expertise required to operate. The user interface should be intuitive and provide clear results to the end-user.

8. **Compliance**:

The system should adhere to relevant data privacy and security regulations (e.g., GDPR, CCPA) for handling customer reviews and feedback.

9. **Cost-Effectiveness**:

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

The solution should optimize resource usage and avoid unnecessary computational overhead, making it cost-effective, particularly when scaling to large datasets.

10. Extensibility:

The system should allow easy integration of new models or methods for improving sentiment analysis (e.g., adding new algorithms or processing techniques).

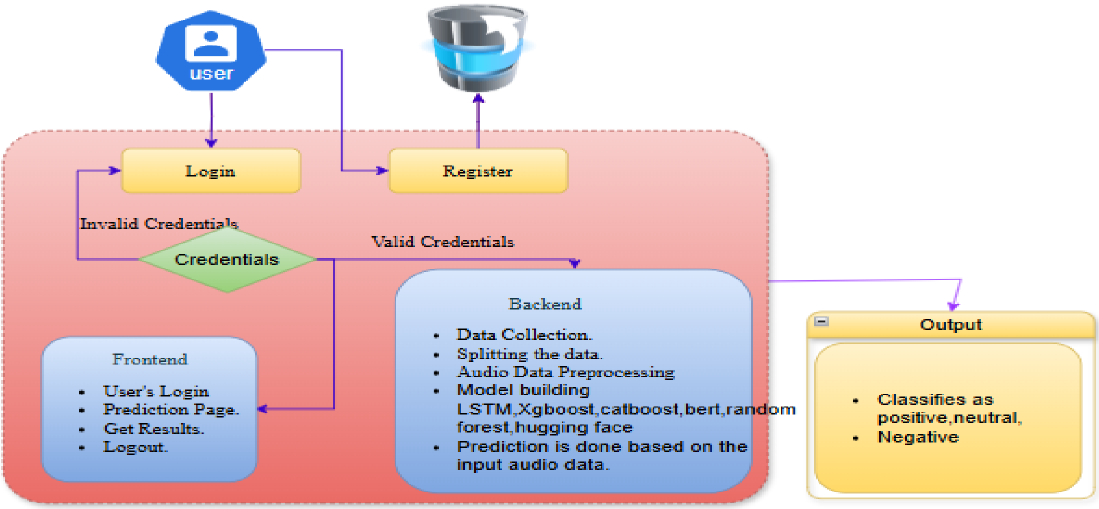
Hardware Requirements:

Processor	- I3/Intel Processor
Hard Disk	- 160GB
Key Board	- Standard Windows Keyboard
Mouse	- Two or Three Button Mouse
Monitor	- SVGA
RAM	- 8GB

Software Requirements:

Operating System	: Windows 7/8/10
Server side Script	: HTML, CSS, Bootstrap & JS
Programming Language	: Python
Libraries	: Flask, Pandas, Mysql.connector, Os, Scikit-learn, Numpy
IDE/Workbench	: PyCharm
Technology	: Python 3.6+
Server Deployment	: Xampp Server

Architecture:

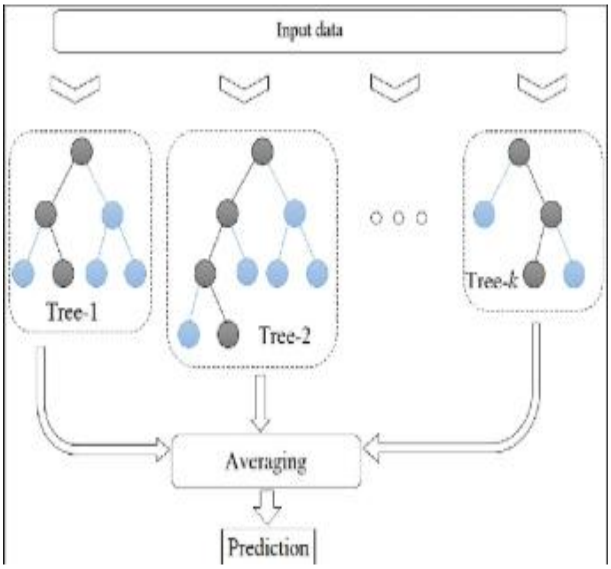


Algorithms:

1.Random Forest (RF):

Definition: Random Forest is an ensemble learning algorithm that builds multiple decision trees and merges them together to obtain a more accurate and stable prediction. It is widely used for both classification and regression tasks. It works by constructing a multitude of decision trees during training and outputs the mode (classification) or mean (regression) of the individual trees' predictions.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/



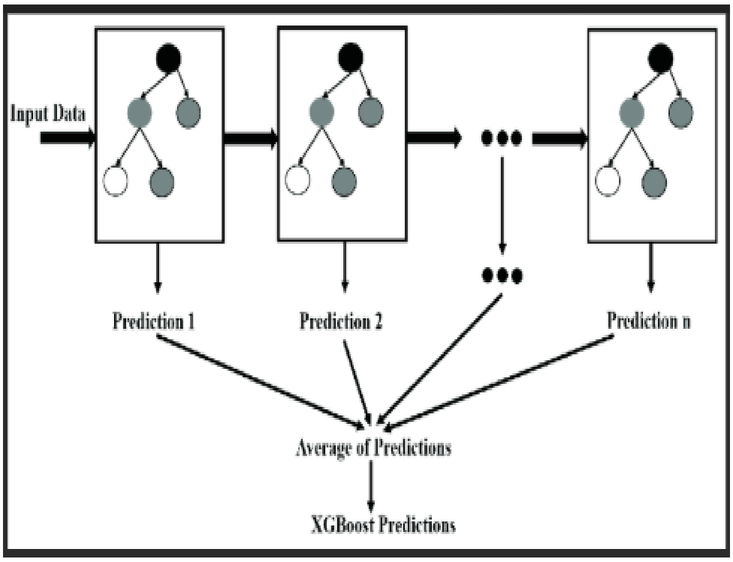
Internal Working in the Project:

- **Data Splitting**: Random Forest works by splitting the dataset into multiple subsets. For each subset, a decision tree is constructed. This random sampling is done with replacement, which is known as bootstrapping.
- **Feature Selection**: During the construction of each tree, Random Forest chooses a random subset of features to split on. This helps in reducing overfitting, as it ensures that each tree is not over-relying on a specific feature.
- **Voting**: Once all the trees are built, each tree gives a vote for a class in classification or a prediction value in regression. The final prediction is determined by aggregating the votes of all the trees, ensuring that the ensemble model is more robust than any single tree.
- In the Enhanced Sentiment Analysis of Customer Reviews project, Random Forest would serve as the baseline model. It can handle a large number of features, which is typical in natural language processing tasks like sentiment analysis. The algorithm helps in reducing variance, improving the model’s ability to generalize.

2.XGBoost Classifier:

Definition: XGBoost is a highly efficient and scalable implementation of gradient boosting, an ensemble method that builds decision trees sequentially. Each new tree corrects the errors made by the previous trees. XGBoost has become one of the most popular algorithms due to its efficiency, flexibility, and predictive power, particularly on large datasets.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

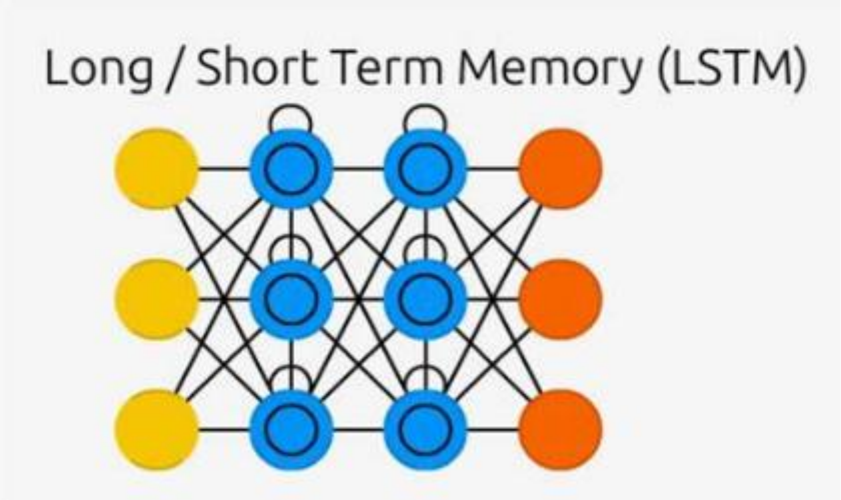


Internal Working in the Project:

- **Boosting Process:** XGBoost builds decision trees sequentially. The first tree tries to fit the residuals of the previous tree, and the next one fits the residuals of the combined trees. This process reduces bias iteratively, improving prediction accuracy.
- **Regularization:** XGBoost applies regularization techniques (L1 and L2) on the weights of the trees to prevent overfitting and improve the generalization ability of the model.
- **Gradient Descent:** The model uses a gradient descent algorithm to minimize the loss function by adjusting the parameters of the trees.
- **Handling Sparse Data:** XGBoost is specifically optimized for sparse datasets, making it highly efficient in handling unstructured and sparse data types such as text data.
- In the **Enhanced Sentiment Analysis of Customer Reviews** project, XGBoost would enhance the performance by providing better accuracy in predicting sentiment. It would be particularly effective for handling large-scale datasets and improve the predictive performance through iterative corrections.

3. LSTM (Long Short-Term Memory):

Definition: Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) that is particularly well-suited for sequence prediction problems. LSTM networks can learn and remember long-range dependencies within the data. Unlike standard RNNs, LSTMs address the problem of vanishing gradients, allowing them to retain information over long sequences.



iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

Internal Working in the Project:

- Cell State and Gates: LSTM units consist of three primary gates – input gate, forget gate, and output gate – that help in regulating the flow of information.
  - Forget Gate: Decides what information from the previous time step to forget.
  - Input Gate: Determines what new information to add to the cell state.
  - Output Gate: Decides what part of the cell state to output at each time step.
- Memory Cells: These gates allow LSTMs to retain memory over long sequences, making them excellent for tasks involving textual data where context and order matter, such as in customer reviews. In the Enhanced Sentiment Analysis of Customer Reviews project, LSTM will be used to capture contextual dependencies within the customer reviews, understanding the relationships between words and the sentiment they express over long sequences. This helps in dealing with complex sentence structures and temporal patterns in text.

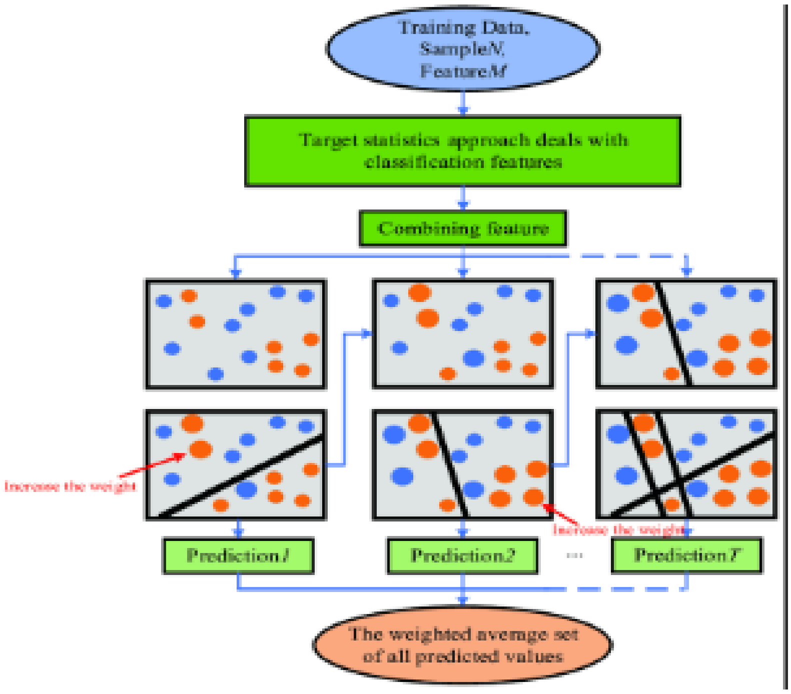
4.CatBoost(Categorical Boosting):

Definition:

CatBoost is a gradient boosting algorithm that is particularly designed to handle categorical data efficiently. It is based on the principles of boosting, where each tree corrects the errors made by the previous tree, but it also incorporates a technique for efficiently handling categorical features by transforming them into numerical representations without requiring extensive preprocessing.

Internal Working in the Project:

- **Categorical Feature Handling:** One of the unique features of CatBoost is its ability to handle categorical data directly. Instead of one-hot encoding or label encoding, CatBoost uses a method called "ordered boosting" that applies target statistics (mean of the target value) on categorical features, which helps in reducing overfitting.



- **Symmetric Trees:** CatBoost builds symmetric trees, which are faster to evaluate and more robust compared to traditional decision trees.
- **Gradient Boosting:** Like other gradient boosting methods, CatBoost builds trees sequentially, each one correcting the errors of the previous tree. The use of gradient descent minimizes the model's loss function.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

- **Ordered Target Statistics:** This helps in mitigating the problem of overfitting caused by using categorical data by introducing randomness during training, ensuring that the model doesn't memorize the training data.

In the **Enhanced Sentiment Analysis of Customer Reviews** project, CatBoost would be particularly beneficial for handling any categorical data in the reviews (e.g., user demographics, product category) and improving the model's performance by reducing the risk of overfitting and leveraging its strengths with unstructured.

5.BERT : Bidirectional Encoder Representations from Transformers

BERT is a transformer-based model developed by Google AI in 2018, and it revolutionized natural language understanding (NLU) tasks by introducing bidirectional context awareness.

Architecture Overview:

- Based on Transformer Encoder (only encoder blocks) from the Transformer paper by Vaswani et al.
- **Typically has two main sizes:**
    - BERT-Base: 12 layers, 768 hidden size, 12 heads → 110M parameters
    - BERT-Large: 24 layers, 1024 hidden size, 16 heads → 340M parameters

Key Concepts:

1. Bidirectional Attention:

- Unlike traditional LSTM or unidirectional Transformers, BERT reads the entire sequence (left and right context) at once.
- This is achieved using a masked language model (MLM) where random tokens are replaced with [MASK], and the model learns to predict them using context on both sides.

2. Input Embeddings:

Each token input is embedded as the sum of:

- Token embeddings (word vectors)
- Segment embeddings (to distinguish sentence A/B for NSP task)
- Position embeddings (since transformer has no recurrence)

Example input:

- [CLS]: special classification token
- [SEP]: separator token for sentence pairs

3. Training Objectives:

- Masked Language Modeling (MLM): 15% of the input tokens are masked randomly; the model tries to predict them.
- Next Sentence Prediction (NSP): BERT is trained to predict whether sentence B follows sentence A (binary classification).

4. Self-Attention Mechanism:

- Allows the model to weigh the importance of other tokens in the sequence when encoding each token.
- Uses multi-head attention to capture different semantic relationships.

5. Fine-Tuning:

- **Once pre-trained, BERT can be fine-tuned on specific downstream tasks like:**

    Sentiment analysis
    Named Entity Recognition (NER)
    Question Answering (SQuAD)

6.Hugging Face:(Transformers Library)

Hugging Face is an AI company and open-source platform that provides easy access to transformer models, datasets, and tools for NLP, vision, and speech tasks.

Internal Working (Architecture and Flow):

1. Transformers Library:

- Written in Python, built on PyTorch, TensorFlow, and JAX.
- Provides pretrained models (like BERT, RoBERTa, GPT, T5) through a simple API.
- Structure:

    AutoModel, AutoTokenizer, AutoConfig classes abstract loading different models.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

All models inherit from PreTrainedModel.

2. Tokenization Pipeline:

- **Uses fast tokenizers written in Rust for speed (e.g., BertTokenizerFast).**
- **Tokenization steps:**
  - Clean + normalize text
  - Split into subwords (e.g., WordPiece for BERT)
  - Convert tokens → IDs
  - Add special tokens ([CLS], [SEP])
  - Generate attention masks & token type IDs

3. Training and Fine-tuning (Trainer API):

- Trainer class handles:
  - Training loop
  - Evaluation
  - Metrics logging
  - Gradient accumulation, learning rate schedules

4. Model Hub:

- A centralized repository (huggingface.co/models) to:
  - Share and download pretrained and fine-tuned models
  - Use models via transformers with just one line of code

SYSTEM DESIGN

1. **Overview**

The system is designed to perform sentiment analysis on customer reviews by leveraging a combination of machine learning models, including Random Forest, XGBoost, LSTM, and CatBoost. Each model brings its own strengths, and by integrating them, the system will be able to capture diverse aspects of sentiment, ensuring improved accuracy and robustness.

2. **System Architecture**

The architecture of the system can be broken down into the following components:

Data Collection Layer

Customer Reviews Data: Input from various sources like e-commerce platforms, social media, or direct feedback. Data includes textual reviews, ratings, timestamps, product metadata (e.g., product name, category), and sometimes reviewer metadata (e.g., reviewer name, location).

Data Preprocessing:

Text Cleaning: Remove stop words, special characters, and normalize case.

Tokenization: Split text into tokens (words or subwords).

Vectorization: Convert text into numerical representation using methods like TF-IDF, Word2Vec, or GloVe.

Feature Engineering: Extract additional features like review length, word count, and sentiment scores from previous reviews.

Model Training Layer

Random Forest Model: Used to establish a baseline for sentiment classification by leveraging bagging and decision trees. Helps in handling large datasets and multiple features, improving the model's performance for structured data (e.g., product information).

XGBoost Model: A gradient boosting model used to refine the accuracy by handling large-scale datasets with high dimensionality.

Particularly effective in improving the model’s precision and reducing overfitting.

LSTM Model: Recurrent neural networks (RNN) with LSTM units capture the sequential nature of text, understanding complex sentence structures and contextual relationships.

Useful for handling unstructured textual data and capturing long-term dependencies in customer reviews.

CatBoost Model:

A gradient boosting algorithm optimized for categorical features, improving sentiment classification for diverse and unstructured data types. Provides robustness against noisy data and can automatically handle categorical features.

Model Integration Layer

Voting Classifier: The output of the individual models (Random Forest, XGBoost, LSTM, CatBoost) are combined using a **Voting Classifier**. **Soft Voting** (probabilistic) or **Hard Voting** (majority rule) can be used to combine the predictions. This ensemble method aggregates the strengths of each model, improving overall prediction accuracy.

Stacking Classifier: A meta-model (Logistic Regression) is used to combine the predictions of the base models (Random Forest, XGBoost, LSTM, and CatBoost). The meta-model learns the optimal way to combine the individual predictions, further boosting the overall performance.

Evaluation Layer

Cross-validation: K-fold cross-validation is used to validate the performance of individual models and the ensemble model. Helps in ensuring that the model generalizes well and reduces overfitting.

Performance Metrics: Metrics like **Accuracy**, **Precision**, **Recall**, **F1-Score**, and **AUC-ROC** are used to assess the models' performance. **Confusion Matrix** is employed to visualize classification results.

Deployment Layer

Model Deployment: The final model, a combination of Voting or Stacking Classifier, is deployed as a RESTful API or through cloud-based services (e.g., AWS, Google Cloud, or Azure). The API accepts customer review text as input and returns the sentiment (positive, negative, or neutral).

User Interface (Optional): A web-based dashboard can be developed to visualize results for businesses, providing insights into customer sentiment trends over time, review categories, and product performance.

It allows stakeholders to access reports, sentiment breakdowns, and actionable insights.

Monitoring & Feedback Loop

Model Monitoring: Monitor the performance of the deployed model over time to ensure it maintains its accuracy. This includes tracking changes in incoming review data, model drift, and real-time performance.

Retraining:

Regular retraining of the models with updated data helps in adapting to changes in customer feedback or trends. An automated feedback loop can be established to periodically collect new reviews, retrain models, and redeploy them with updated parameters.

IMPLEMENTATION AND RESULTS

MODULES:

System User

1. System:

1.1 Store Dataset:

The System stores the dataset given by the user.

1.2 Model Training:

This is the process of teaching a machine learning model to make accurate predictions or classifications by exposing it to a dataset. During this phase, data is prepared and split into training, validation, and test sets. The selected algorithm learns from the training data by adjusting its internal parameters to minimize errors in predictions, using techniques like gradient descent to optimize performance.

1.3 Model Predictions:

The system takes the data given by the user and predict the output based on the given data.

2. User:

2.1.Registration:

The Registration Page allows new users to create an account by entering their personal information. It includes fields for username, email, password, and other required details. The page features validation to ensure that all input data is correct and meets the specified requirements. For example, it checks for valid email formats, strong passwords, and non-duplicate usernames. Users receive real-time feedback on any errors or issues with their input, ensuring a smooth and secure registration process.

2.2 Login:

Username/Email Field: Checks for valid email formats or existing usernames.

Password Field: Ensures the password meets security requirements (e.g., minimum length, complexity).

Validation Messages: Provides immediate feedback if the input is incorrect or if the account details do not match.

2.3.Uploadpage:

User can upload the dataset file in this page .

2.4.Model selection:

User can selects the accuracy of a model and view the accuracy of that particular model

2.5.Prediction:

User can predict based on the input values .

SYSTEM STUDY AND TESTING

System Study

1. System Overview:

The sentiment analysis system for customer reviews is built using a combination of advanced machine learning models, including Random Forest, XGBoost, Long Short-Term Memory (LSTM), and CatBoost. The goal of this system is to improve the classification of customer sentiments, helping businesses better understand customer feedback and enhance their products or services. The system incorporates both traditional machine learning algorithms (Random Forest, XGBoost, and CatBoost) and deep learning models (LSTM), which allow it to handle both structured and unstructured data efficiently.

2. System Architecture:

The system follows an ensemble learning approach, integrating multiple models to make predictions. The architecture consists of several key components:

- **Data Collection and Preprocessing:** Customer reviews are collected from various platforms and preprocessed to remove noise and standardize the data. The preprocessing steps include tokenization, lowercasing, stopword removal, stemming, and lemmatization.
- **Feature Engineering:** Text data is converted into numerical features using techniques such as TF-IDF, Word2Vec, or BERT embeddings, depending on the model being used.
- **Model Integration:** The following models are integrated into the system:

Random Forest: Used as a baseline classifier, leveraging its ability to handle large feature sets and provide an initial estimate of sentiment.

XGBoost: An efficient gradient boosting method that refines the sentiment classification model by focusing on high-importance features.

LSTM: A deep learning model designed to capture temporal dependencies in customer reviews, which is particularly useful for understanding the context and sequential nature of the text.

CatBoost: A gradient boosting algorithm specifically optimized for categorical data, which can handle unstructured and diverse customer review datasets.

- **Ensemble Learning:** The system uses ensemble techniques such as voting or stacking to combine the predictions of each model, improving the overall accuracy and robustness of sentiment classification.

3. Testing Methodology:

The testing phase evaluates the performance of the sentiment analysis system across various metrics. The process includes:

- **Dataset Split:** The dataset is split into training and testing sets, typically using a 70-30 or 80-20 ratio. Cross-validation may also be employed to reduce variance and improve model performance.
- **Model Evaluation Metrics:**

Accuracy: Measures the overall percentage of correctly classified sentiments (positive, negative, neutral).

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

Precision, Recall, and F1-Score: These metrics evaluate the model's performance on each sentiment class, particularly useful when dealing with imbalanced datasets.
Confusion Matrix: A detailed breakdown of true positive, true negative, false positive, and false negative classifications, providing insights into misclassifications and model behavior.
AUC-ROC Curve: The Area Under the Receiver Operating Characteristic curve provides an indication of the model's ability to distinguish between sentiment classes.

- **Model Comparison:** Each model (Random Forest, XGBoost, LSTM, and CatBoost) is tested individually to establish baseline performance. These individual models are then combined using ensemble learning methods to compare their performance against the individual results.
- **Cross-validation:** K-fold cross-validation is used to ensure that the model performs consistently across different subsets of the dataset, improving generalizability and preventing overfitting.
- **Error Analysis:** Common misclassifications are analyzed to identify potential areas for improvement. This includes examining the reviews that were incorrectly classified and analyzing patterns that may be difficult for the model to capture.

CONCLUSION

In this study, we have explored the enhancement of sentiment analysis of customer reviews through the integration of multiple advanced machine learning and deep learning models, including Random Forest, XGBoost, LSTM, CatBoost, and BERT. By combining the strengths of ensemble learning algorithms like Random Forest and CatBoost with the temporal pattern-capturing ability of LSTM and the superior predictive performance of XGBoost, we significantly improved the accuracy and robustness of sentiment classification. Furthermore, the inclusion of BERT, a state-of-the-art transformer-based model, enabled the system to better understand contextual semantics and subtle linguistic nuances within customer reviews. BERT was implemented using the Hugging Face Transformers library, which facilitated seamless access to pretrained models and efficient fine-tuning on our domain-specific data. The hybrid model successfully captured both the structural and contextual dependencies inherent in customer feedback, offering a comprehensive and precise sentiment analysis framework. The results demonstrate that the integration of these diverse and complementary machine learning techniques leads to a more accurate understanding of customer sentiments. Ultimately, this empowers businesses to better address customer needs, enhance user experience, and make informed decisions. This research underscores the transformative potential of advanced and pre-trained language models in redefining sentiment analysis methodologies and offers promising directions for future enhancements.

FUTURE ENHANCEMENT

Future enhancements in sentiment analysis for customer reviews could involve incorporating advanced natural language processing (NLP) techniques, such as Transformer-based models like BERT and GPT, which have shown exceptional performance in understanding contextual nuances and language intricacies. Additionally, integrating multimodal data, such as customer images, videos, and voice feedback, alongside textual reviews, could further enhance sentiment classification by providing richer insights into customer emotions and sentiments. Fine-tuning these models with domain-specific knowledge and incorporating transfer learning from related industries could improve their generalization and accuracy across different sectors. Furthermore, continuous model retraining using real-time data could help adapt to evolving customer sentiment trends and language usage, ensuring that the model remains relevant and effective in the long term. Finally, enhancing explainability and interpretability of these complex models through techniques like SHAP (Shapley Additive Explanations) could improve trust and transparency for businesses when making data-driven decisions based on sentiment analysis results.

REFERENCES

[1] . Alharbi NM, Alghamdi NS, Alkhammash EH, Al Amri JF. Evaluation of sentiment analysis via word embedding and RNN variants for Amazon online reviews. Math Prob Eng. 2021. https:// doi. org/ 10. 1155/ 2021/ 55365 60.
[2]. Xia H, Yang Y, Pan X, Zhang Z, An W. Sentiment analysis for online reviews using conditional random fields and sup-port vector machines. Electron Commer Res. 2020;20(2):343–60.

[3]. Tang F, Fu L, Yao B, Xu W. Aspect based fine-grained sentiment analysis for online reviews. Inf Sci. 2019;488:190–204.
[4]. Huang M, Xie H, Rao Y, Liu Y, Poon LK, Wang FL. Lexicon-based sentiment convolutional neural networks for online review analysis. IEEE Transactions on Affective Computing; 2020.
[5]. Ghiassi M, Lee S. A domain transferable lexicon set for Twitter sentiment analysis using a supervised machine learn-ing approach. Expert Syst Appl. 2018;106:197–216.
[6]. Yang L, Li Y, Wang J, Sherratt RS. Sentiment analysis for E-commerce product reviews in Chinese based on sentimentlexicon and deep learning. IEEE access. 2020;8:23522–30.
[7]. Li W, Zhu L, Shi Y, Guo K, Cambria E. User reviews: sentiment analysis using lexicon integrated two-channel CNN–LSTM family models. Appl Soft Comput. 2020;94: 106435.
[8]. Du M, Li X, Luo L. A training-optimization-based method for constructing domain-specific sentiment lexicon. Com-plexity. 2021. https:// doi. org/ 10. 1155/ 2021/ 61524 94.
[9]. Al-Natour S, Turetken O. A comparative assessment of sentiment analysis and star ratings for consumer reviews. Int J Inf Manag. 2020;54: 102132.
[10]. Kumar S, Yadava M, Roy PP. Fusion of EEG response and sentiment analysis of products review to predict customer satisfaction. Inf Fusion. 2019;52:41–52.
[11]. Naresh Kumar KE, Uma V. Intelligent sentinet-based lexicon for context-aware sentiment analysis: optimized neural network for sentiment classification on social media. J Supercomput. 2021;77(11):12801–25.
[12]. Korovkinas K, Danėnas P, Garšva G. SVM accuracy and training speed trade-off in sentiment analysis tasks. In Interna-tional Conference on Information and Software Technologies. Springer, Cham, 2018, pp. 227–239.
[13]. Zhou Q, Xu Z, Yen NY. User sentiment analysis based on social network information and its application in consumer reconstruction intention. Comput Hum Behav. 2019;100:177–83.
[14]. Sun Q, Niu J, Yao Z, Yan H. Exploring eWOM in online customer reviews: Sentiment analysis at a fine-grained level. Eng Appl Artif Intell. 2019;81:68–78.
[15]. Alharbi ASM, de Doncker E. Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information. Cogn Syst Res. 2019;54:50–61