# SENTIMENT ANALYSIS FOR PRODUCT REVIEWS

**Shreyas Allani**

**Date: 26-09-2023**

***Abstract:***

In the modern age of e-commerce, understanding customer sentiment is crucial for businesses to make informed decisions and enhance user experience. This report details the development of a sentiment analysis system for product reviews and the creation of a user-friendly web application for real-time sentiment analysis.

## 1. Problem Statement

E-commerce businesses receive a vast number of product reviews from customers. Analyzing these reviews manually is time-consuming and often impractical. The problem statement is to develop an automated sentiment analysis system that can process and classify product reviews into positive, negative, or neutral sentiments .

## 2. Market/Customer Need Assessment

E-commerce platforms, online marketplaces, and retailers can greatly benefit from this sentiment analysis system. The need arises from the following considerations:

- Efficiently process a large volume of customer reviews.
- Understand customer satisfaction and identify areas for improvement.
- Tailor marketing and product development strategies based on customer feedback.

## 3. Target Specification and characterization

The target specifications for this project include:

- Implementing a sentiment analysis model.
- Developing a user-friendly web interface for sentiment analysis.
- Classifying reviews into three categories: positive, negative, or neutral.
- Providing users with real-time sentiment analysis results.

## 4. External Search(information sources)

The dataset used for training and testing the sentiment analysis model is sourced from various e-commerce platforms and includes text reviews paired with their sentiment labels (positive, negative, or neutral). Popular sources of similar datasets include Amazon product reviews, Yelp reviews, and Kaggle datasets.

Text Processing Libraries:

External libraries and tools play a crucial role in text preprocessing and feature extraction. We utilize popular libraries like NLTK (Natural Language Toolkit) and spaCy for tasks such as tokenization, stemming, and part-of-speech tagging.

Machine Learning and Deep Learning Frameworks:

For model development and training, we harness machine learning and deep learning frameworks, including:

Scikit-Learn: We use Scikit-Learn to implement traditional machine learning models such as Logistic Regression and Support Vector Machines for baseline sentiment analysis.

TensorFlow and Keras: These frameworks enable us to build and train deep learning models, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), for more complex sentiment analysis tasks.

## 5. Benchmarking

1. Model Selection:

We consider a range of sentiment analysis models, including both traditional machine learning models and deep learning models:

Machine Learning-Based Models: These models involve training on labelled datasets and can capture nuanced sentiments. Common algorithms include Logistic Regression, Naive Bayes, Support Vector Machines, and Random Forest.

Deep Learning Models: We explore deep learning architectures such as Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Transformer-based models like BERT. Deep learning models have the potential to capture intricate sentiment patterns in text data.

2. Evaluation Metrics:

These metrics include:

Accuracy: Measures the proportion of correctly classified reviews.

Precision: Reflects the model's ability to correctly identify positive, negative, and neutral reviews.

Recall: Measures the model's ability to capture all relevant reviews of a specific sentiment.

F1-Score: The harmonic mean of precision and recall, providing a balance between the two.

3. Datasets:

We use publicly available sentiment analysis datasets related to product reviews. Common sources include:

Amazon Product Reviews Dataset: This dataset contains user reviews of various products from Amazon, with labeled sentiments (positive, negative, neutral).

4. Cross-Validation:

To ensure reliable results, we employ cross-validation techniques helps prevent overfitting and provides a better estimate of the models' performance.

5. Hyperparameter Tuning:

For machine learning-based models and deep learning models, we perform hyperparameter tuning to optimize model parameters, improving overall performance.

## 6. Applicable Patents

Sentiment analysis on IMDB using lexicon and neural network Research Article Published: 02 January 2020 volume 2, Article number: 148 (2020)

U.S. application Ser. No. 11/739,187, filed Apr. 24, 2007, now U.S. Pat. No. 7,996,21

## 7. Applicable Regulations

To ensure ethical and responsible data usage, the following regulations and practices are adhered to:

- Compliance with data privacy laws.
- Transparency in data collection and usage.
- User consent for data processing.
- Protection of sensitive information in reviews.

## 8. Applicable Constraints:

Constraints associated with this project include:

- The need for continuous model retraining with new data.
- Limited computational resources for real-time analysis.
- Ensuring user-friendliness in the web application interface.

## 9. Business Opportunity

The sentiment analysis system can be monetized through the following opportunities:

- Licensing the sentiment analysis API to e-commerce platforms.
- Offering sentiment analysis as a service to businesses.
- Integration with customer relationship management (CRM) systems.

## 10. Concept Development

The concept for this project involves:

- Data preprocessing and cleaning.
- Training and fine-tuning a sentiment analysis model.
- Developing a web application for user interaction.
- Real-time analysis of user-provided text reviews.

## 11. Final Report Prototype

The product takes the following functions to perfect and provide a good result.

**Back-end**

- Data Preprocessing: Clean and preprocess the review text data, including text normalization, tokenization, and feature extraction.
- Model Development: Train a sentiment analysis model using popular libraries for natural language processing.
- Real-time Analysis: Implement a server or serverless function to perform real-time sentiment analysis on user input.

**Front End**

- User Interface: Create an intuitive web interface where users can input text reviews.
- Real-time Feedback: Display sentiment analysis results in real-time, categorized as positive, negative, or neutral.
- Feedback Mechanism: Allow users to provide feedback on the accuracy of sentiment analysis results.

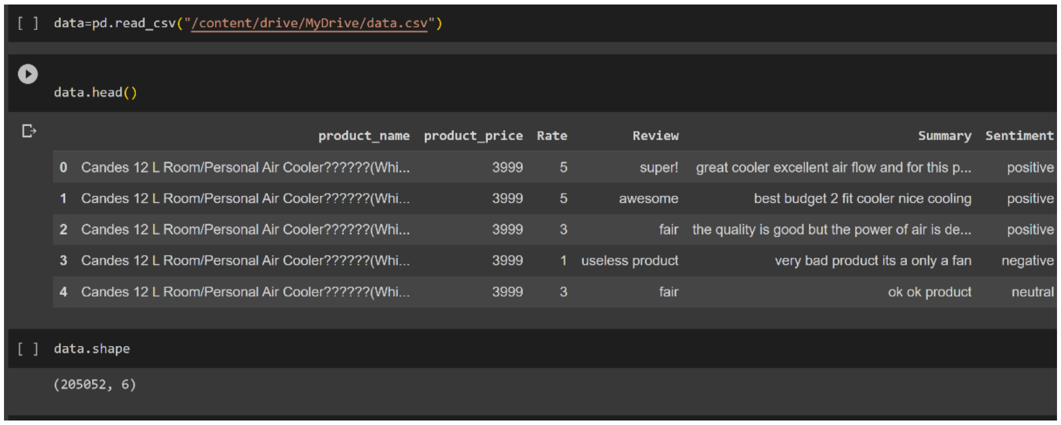
## 12. Product details - How does it work?

The product workflow is as follows:

- Users access the web application.
- Users input a text review.
- The application sends the text to the back-end for analysis.
- The sentiment analysis model classifies the sentiment as positive, negative, or neutral.
- The result is displayed to the user in real-time.
-

## 13. Code Implementation

- Accessing Dataset and Reading the Meta Data



```
data=pd.read_csv("/content/drive/MyDrive/data.csv")
```

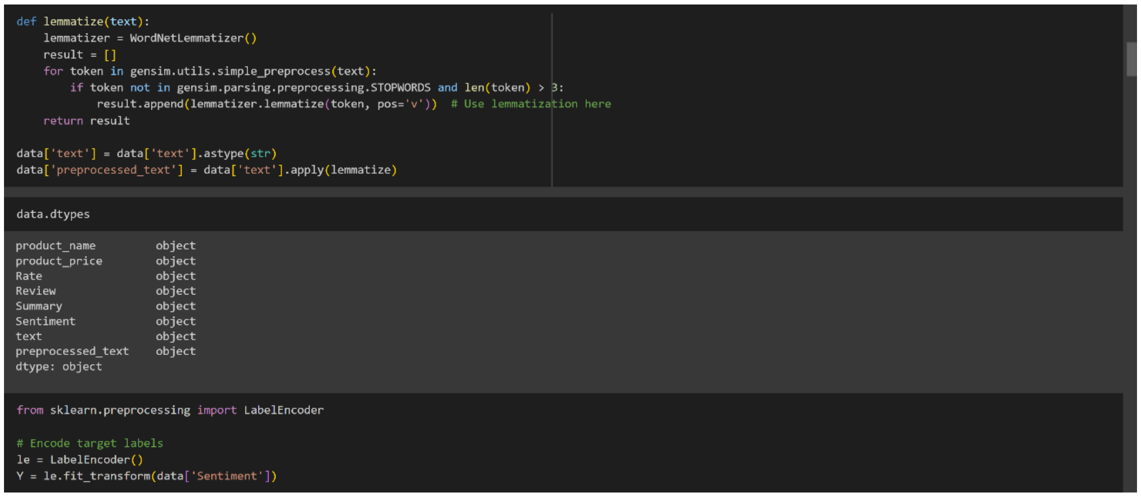
```
data.head()
```

```
                                   product_name  product_price  Rate          Review                                        Summary  Sentiment
0  Candes 12 L Room/Personal Air Cooler??????(Whi...       3999     5          super!  great cooler excellent air flow and for this p...   positive
1  Candes 12 L Room/Personal Air Cooler??????(Whi...       3999     5         awesome              best budget 2 fit cooler nice cooling   positive
2  Candes 12 L Room/Personal Air Cooler??????(Whi...       3999     3            fair  the quality is good but the power of air is de...   positive
3  Candes 12 L Room/Personal Air Cooler??????(Whi...       3999     1  useless product                  very bad product its a only a fan   negative
4  Candes 12 L Room/Personal Air Cooler??????(Whi...       3999     3            fair                                     ok ok product    neutral
```

```
data.shape
```

```
(205052, 6)
```

- Dataset Preprocessing using techniques like Stemming, Lemmatizeing  and removing stopwords



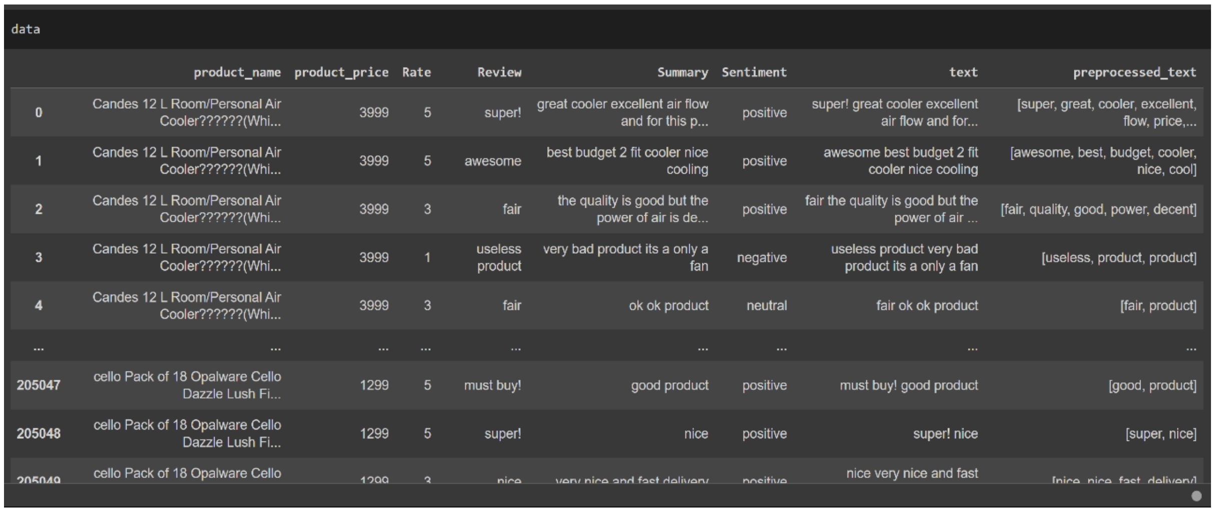
```
def lemmatize(text):
    lemmatizer = WordNetLemmatizer()
    result = []
    for token in gensim.utils.simple_preprocess(text):
        if token not in gensim.parsing.preprocessing.STOPWORDS and len(token) > 3:
            result.append(lemmatizer.lemmatize(token, pos='v'))  # Use lemmatization here
    return result

data['text'] = data['text'].astype(str)
data['preprocessed_text'] = data['text'].apply(lemmatize)
```

```
data.dtypes
```

```
product_name         object
product_price        object
Rate                 object
Review               object
Summary              object
Sentiment            object
text                 object
preprocessed_text    object
dtype: object
```

```
from sklearn.preprocessing import LabelEncoder

# Encode target labels
le = LabelEncoder()
Y = le.fit_transform(data['Sentiment'])
```

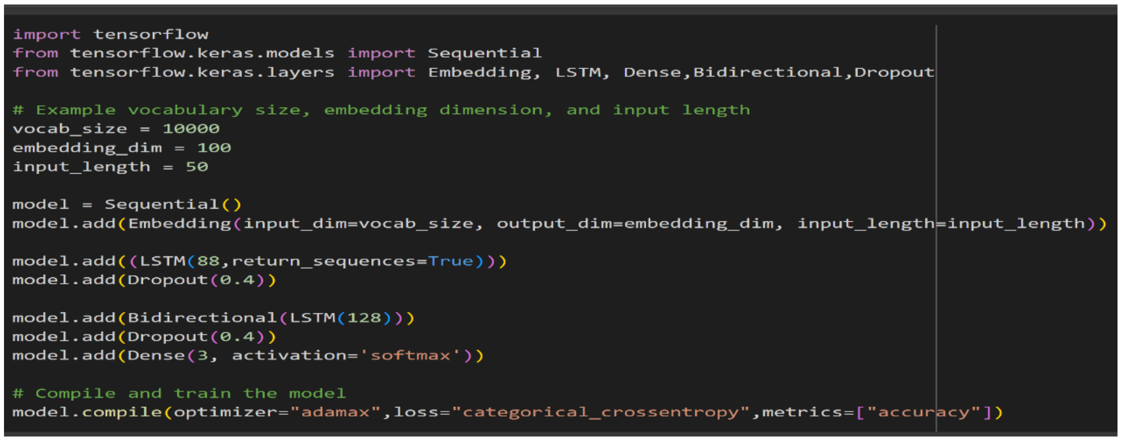
After Applying the Function



```
data
```

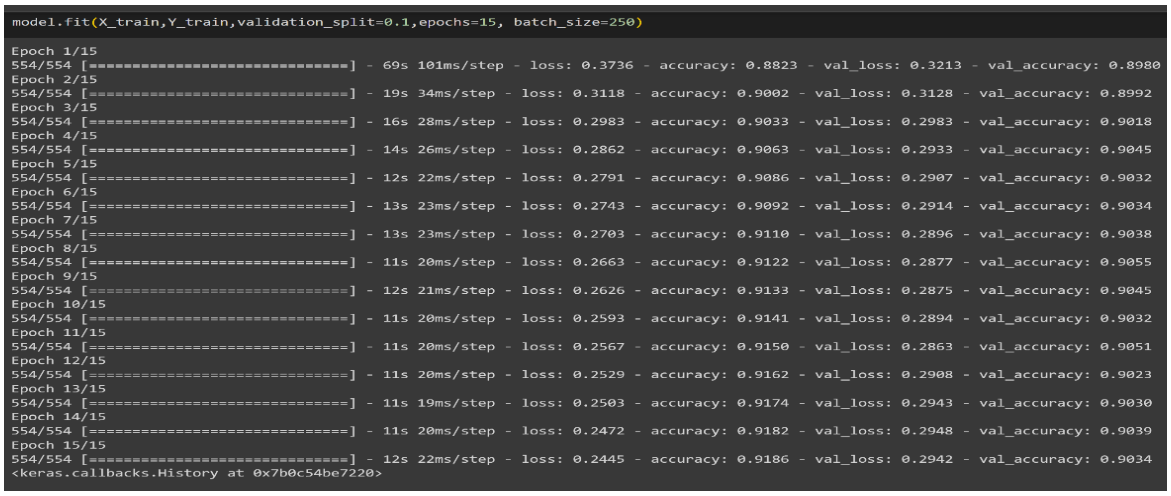
| | product_name | product_price | Rate | Review | Summary | Sentiment | text | preprocessed_text |
|---|---|---|---|---|---|---|---|---|
| **0** | Candes 12 L Room/Personal Air Cooler??????(Whi... | 3999 | 5 | super! | great cooler excellent air flow and for this p... | positive | super! great cooler excellent air flow and for... | [super, great, cooler, excellent, flow, price,... |
| **1** | Candes 12 L Room/Personal Air Cooler??????(Whi... | 3999 | 5 | awesome | best budget 2 fit cooler nice cooling | positive | awesome best budget 2 fit cooler nice cooling | [awesome, best, budget, cooler, nice, cool] |
| **2** | Candes 12 L Room/Personal Air Cooler??????(Whi... | 3999 | 3 | fair | the quality is good but the power of air is de... | positive | fair the quality is good but the power of air ... | [fair, quality, good, power, decent] |
| **3** | Candes 12 L Room/Personal Air Cooler??????(Whi... | 3999 | 1 | useless product | very bad product its a only a fan | negative | useless product very bad product its a only a fan | [useless, product, product] |
| **4** | Candes 12 L Room/Personal Air Cooler??????(Whi... | 3999 | 3 | fair | ok ok product | neutral | fair ok ok product | [fair, product] |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **205047** | cello Pack of 18 Opalware Cello Dazzle Lush Fi... | 1299 | 5 | must buy! | good product | positive | must buy! good product | [good, product] |
| **205048** | cello Pack of 18 Opalware Cello Dazzle Lush Fi... | 1299 | 5 | super! | nice | positive | super! nice | [super, nice] |
| **205049** | cello Pack of 18 Opalware Cello | 1299 | 3 | nice | very nice and fast delivery | positive | nice very nice and fast | [nice, nice, fast, delivery] |

- Model Building using various layers (DL implementation)



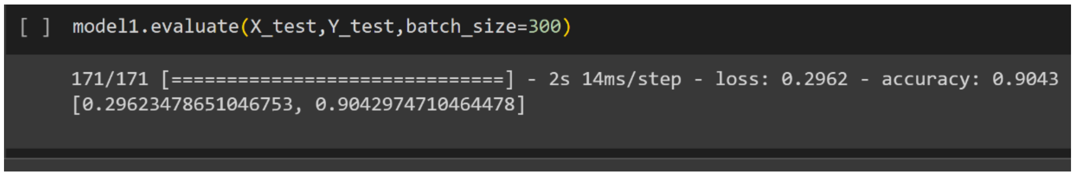
```
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense,Bidirectional,Dropout

# Example vocabulary size, embedding dimension, and input length
vocab_size = 10000
embedding_dim = 100
input_length = 50

model = Sequential()
model.add(Embedding(input_dim=vocab_size, output_dim=embedding_dim, input_length=input_length))

model.add((LSTM(88,return_sequences=True)))
model.add(Dropout(0.4))

model.add(Bidirectional(LSTM(128)))
model.add(Dropout(0.4))
model.add(Dense(3, activation='softmax'))

# Compile and train the model
model.compile(optimizer="adamax",loss="categorical_crossentropy",metrics=["accuracy"])
```

Model Fitting



```
model.fit(X_train,Y_train,validation_split=0.1,epochs=15, batch_size=250)

Epoch 1/15
554/554 [==============================] - 69s 101ms/step - loss: 0.3736 - accuracy: 0.8823 - val_loss: 0.3213 - val_accuracy: 0.8980
Epoch 2/15
554/554 [==============================] - 19s 34ms/step - loss: 0.3118 - accuracy: 0.9002 - val_loss: 0.3128 - val_accuracy: 0.8992
Epoch 3/15
554/554 [==============================] - 16s 28ms/step - loss: 0.2983 - accuracy: 0.9033 - val_loss: 0.2983 - val_accuracy: 0.9018
Epoch 4/15
554/554 [==============================] - 14s 26ms/step - loss: 0.2862 - accuracy: 0.9063 - val_loss: 0.2933 - val_accuracy: 0.9045
Epoch 5/15
554/554 [==============================] - 12s 22ms/step - loss: 0.2791 - accuracy: 0.9086 - val_loss: 0.2907 - val_accuracy: 0.9032
Epoch 6/15
554/554 [==============================] - 13s 23ms/step - loss: 0.2743 - accuracy: 0.9092 - val_loss: 0.2914 - val_accuracy: 0.9034
Epoch 7/15
554/554 [==============================] - 13s 23ms/step - loss: 0.2703 - accuracy: 0.9110 - val_loss: 0.2896 - val_accuracy: 0.9038
Epoch 8/15
554/554 [==============================] - 11s 20ms/step - loss: 0.2663 - accuracy: 0.9122 - val_loss: 0.2877 - val_accuracy: 0.9055
Epoch 9/15
554/554 [==============================] - 12s 21ms/step - loss: 0.2626 - accuracy: 0.9133 - val_loss: 0.2875 - val_accuracy: 0.9045
Epoch 10/15
554/554 [==============================] - 11s 20ms/step - loss: 0.2593 - accuracy: 0.9141 - val_loss: 0.2894 - val_accuracy: 0.9032
Epoch 11/15
554/554 [==============================] - 11s 20ms/step - loss: 0.2567 - accuracy: 0.9150 - val_loss: 0.2863 - val_accuracy: 0.9051
Epoch 12/15
554/554 [==============================] - 11s 20ms/step - loss: 0.2529 - accuracy: 0.9162 - val_loss: 0.2908 - val_accuracy: 0.9023
Epoch 13/15
554/554 [==============================] - 11s 19ms/step - loss: 0.2503 - accuracy: 0.9174 - val_loss: 0.2943 - val_accuracy: 0.9030
Epoch 14/15
554/554 [==============================] - 11s 20ms/step - loss: 0.2472 - accuracy: 0.9182 - val_loss: 0.2948 - val_accuracy: 0.9039
Epoch 15/15
554/554 [==============================] - 12s 22ms/step - loss: 0.2445 - accuracy: 0.9186 - val_loss: 0.2942 - val_accuracy: 0.9034
<keras.callbacks.History at 0x7b0c54be7220>
```

- Analyzing Model Acuuracy and Loss



```
[ ]  model1.evaluate(X_test,Y_test,batch_size=300)

     171/171 [==============================] - 2s 14ms/step - loss: 0.2962 - accuracy: 0.9043
     [0.29623478651046753, 0.9042974710464478]
```

## 14.Conclusion

In the ever-evolving landscape of data-driven decision-making, the development and implementation of our customer revenue prediction system represent a significant leap forward for businesses seeking to optimize their marketing efforts and revenue generation. This report has outlined the essential components, methodologies, and insights gleaned from our project.

The primary objective was to address the critical issue faced by e-commerce platforms and businesses, which is to identify and target customers who have the potential to contribute significantly to their revenue. Through the application of machine learning and deep learning techniques to the Online Shopper's Intention dataset and have successfully developed a predictive model capable of forecasting customer revenue.