Flavors of Feedback: Sentiment Analysis of Amazon Food Reviews

This report explains and shows our thought process of using machine learning and natural language processing techniques on a food reviews dataset.

Srihith Duggi
Abhiram Naredla
Veer Patel
Riya Lodha

I 320D: Text Mining & Natural Language Processing
Dr. Abhijit Mishra
May 2, 2024

Introduction

In today's highly competitive business landscape, customer feedback plays a pivotal role in driving product improvements, enhancing customer satisfaction, and ensuring a business's long-term success. However, manually analyzing vast volumes of unstructured customer reviews can be a daunting task, making it challenging for businesses to extract meaningful insights and make informed decisions. This project aims to address this challenge by developing a Natural Language Processing (NLP) model capable of analyzing large-scale food reviews and extracting valuable insights to aid businesses in their decision-making processes.

Problem

Many businesses struggle to effectively extract meaningful insights from customer reviews due to the unstructured nature of the data and the sheer volume of feedback spread across various platforms. This poses several challenges, including:

- Understanding customer sentiment towards products or services
- Identifying specific areas of strength and weakness in their offerings
- Making informed decisions to enhance product quality, customer service, and overall customer satisfaction
- Remaining competitive by quickly adapting to customer needs and market trends

Goal

The primary goal of this project is to develop an NLP model that can analyze large volumes of food reviews and extract valuable insights to aid business decision-making. With machine learning and natural language processing techniques at its core, this model aims to provide businesses with a comprehensive understanding of their customer base, enable data-driven product decisions, and enhance overall customer satisfaction.

Dataset

For this project, the Amazon Fine Food Reviews dataset was utilized, consisting of over 500,000 reviews spanning more than 10 years. To streamline the analysis, only relevant columns were selected, which includes the "Text" column containing the customer review text and the "Score" column representing the customer rating on a scale of 1 to 5.

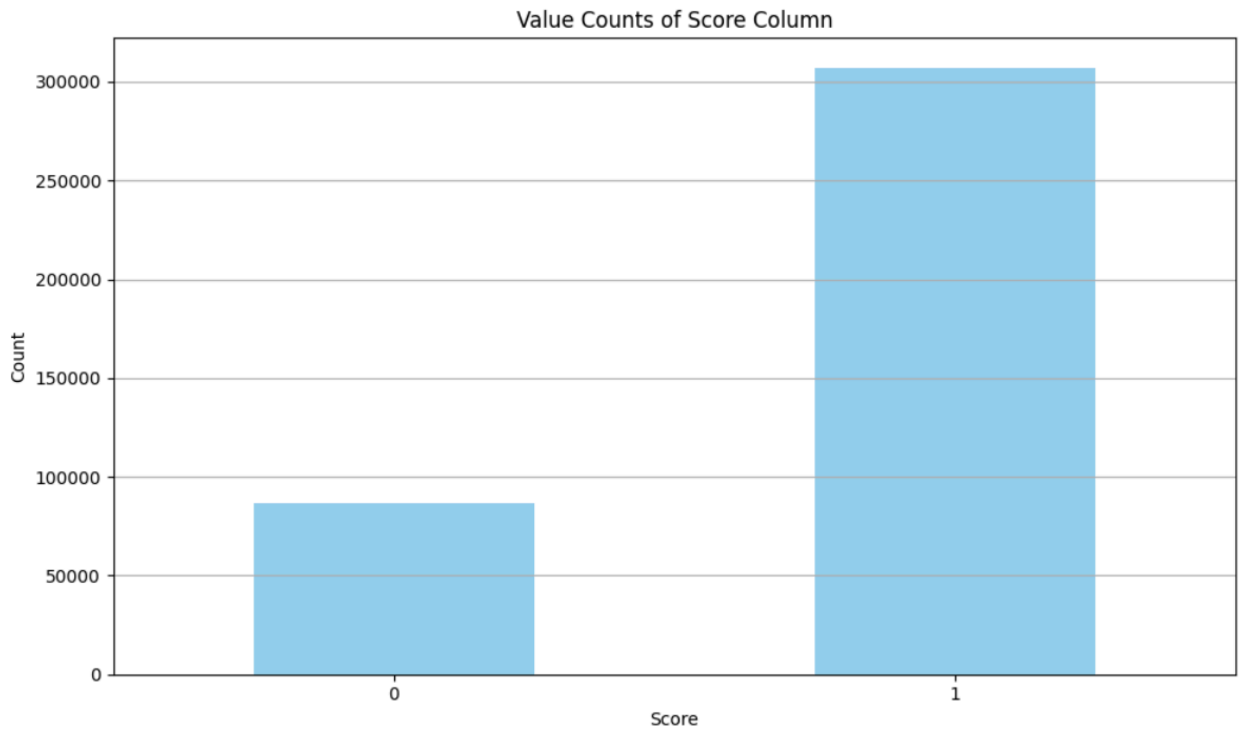
The dataset proved to be a rich source of information for sentiment analysis and topic modeling tasks. To facilitate the analysis, a heuristic was applied to the target variable (Score): reviews with a score of 3 or higher were considered positive (encoded as 1), while those with a score below 3 were classified as negative (encoded as 0).

Data Description

Feature Name	Data Type	Description
Text	Categorical	Text of the review
Score	Numeric	Number denoting sentiment of the Text review - 0 = negative, 1 = positive

<u>Exploratory Data Analysis (EDA)</u>

Before Preprocessing: Imbalanced target variable

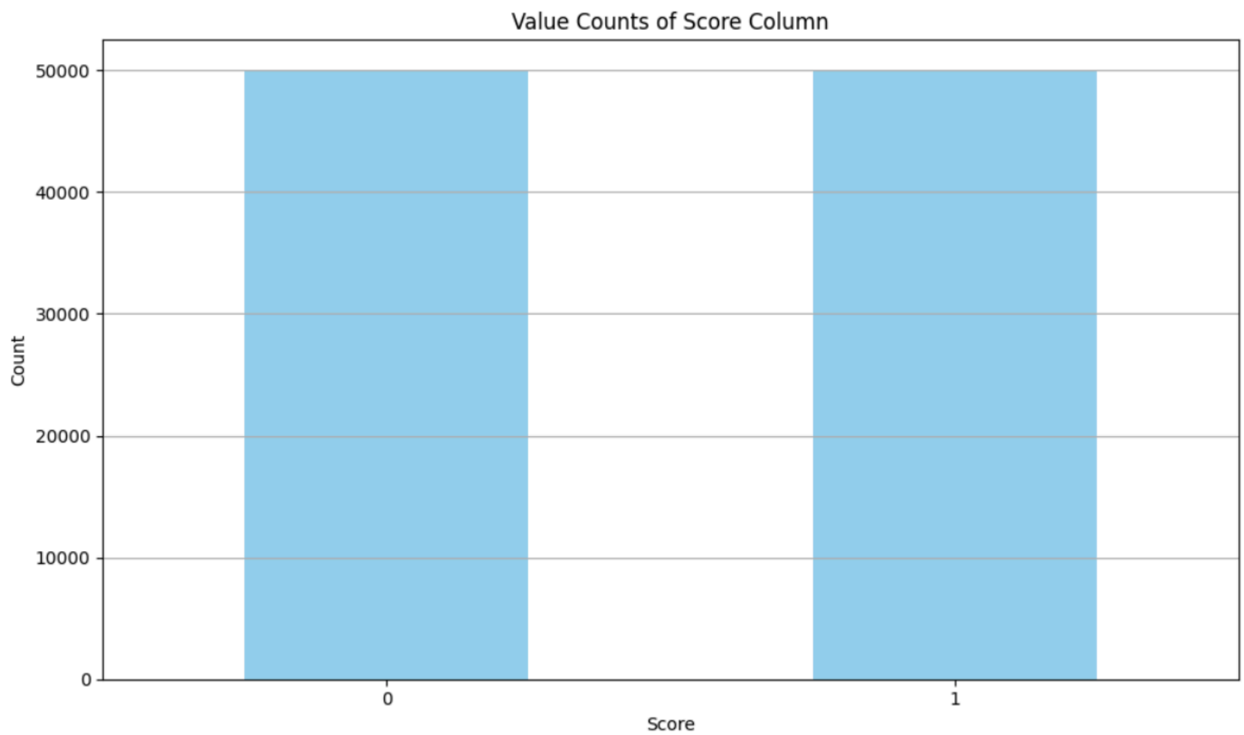


Based on the image above (before preprocessing), it is evident that the Amazon Fine Food Reviews dataset contained a significant imbalance, with a substantially higher number of positive reviews (score = 1) compared to negative reviews (score = 0). To address this imbalance and ensure effective analysis, we took an undersampling approach which involved randomly selecting an equal number of reviews from both classes, effectively balancing the dataset. Specifically, 50,000 reviews were sampled from each class, resulting in a total of 100,000 reviews for analysis.

By undersampling the majority class (positive reviews) and maintaining an equal representation of both positive and negative reviews, the analysis could be conducted on a balanced dataset.

This step was crucial to ensure that the machine learning models or analytical techniques employed would not be unduly influenced by the initially skewed distribution, thereby improving the reliability and accuracy of the subsequent analysis and findings.

After Preprocessing: Balanced target variable



Text Preprocessing

```python
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import re
import string

texts = df_balanced['Text'].astype(str)


def clean_text(text):
    # Convert to lowercase
    text = text.lower()
    # Remove URLs
    text = re.sub(r'http\S+', '', text)
    # Remove HTML tags
    text = re.sub(r'<.*?>', '', text)
    # Remove special characters and digits
    text = re.sub(r'[^\w\s]', '', text)
    # Tokenize the text data
    text = nltk.word_tokenize(text)
    # Remove stopwords
    stop_words = set(stopwords.words('english'))
    text = [word for word in text if word not in stop_words]
    # Apply lemmatization
    lemmatizer = WordNetLemmatizer()
    text = [lemmatizer.lemmatize(word) for word in text]
    # Join the cleaned tokens back into a string
    text = ' '.join(text)
    return text
```

Text preprocessing is a crucial step in natural language processing tasks, including sentiment analysis and topic modeling. It helps in reducing noise, standardizing the data, and focusing on the most informative aspects of the text, ultimately improving the quality and accuracy of the analysis results. Here are some text preprocessing techniques we used in detail:

1. Conversion to Lowercase:

- Purpose: Converting all characters in the text to lowercase helps in standardizing the data and reducing the dimensionality of the feature space.
- Benefits: It ensures that the same words written in different cases (e.g., "Great" and "great") are treated as the same word, preventing the algorithm from considering them as separate features.

2. Removal of URLs:
   - Purpose: URLs are web addresses that typically do not contribute meaningful information to the sentiment analysis or topic modeling tasks. Removing URLs helps in cleaning the text data.
   - Benefits: By eliminating URLs, we focus on the actual content of the reviews and reduce noise in the data.
3. Removal of HTML Tags:
   - Purpose: HTML tags are used to structure content on web pages but do not provide relevant information for text analysis. Removing HTML tags helps in extracting the plain text content from the reviews.
   - Benefits: By stripping away the HTML tags, we obtain clean and readable text data that can be effectively processed by the algorithms.
4. Removal of Special Characters and Digits:
   - Purpose: Special characters (e.g., punctuation marks) and digits often do not carry significant meaning in sentiment analysis or topic modeling. Removing them helps in simplifying the text data.

    - Benefits: By eliminating special characters and digits, we focus on the words and their semantic meaning, reducing the dimensionality of the feature space and improving the efficiency of the algorithms.
5. Tokenization:
    - Purpose: Tokenization is the process of splitting the text into individual words or tokens. It breaks down the text into smaller units that can be processed independently.
    - Benefits: Tokenization allows us to analyze the text at a word level, enabling the extraction of meaningful features and the application of various text analysis techniques.
6. Removal of Stop Words:
    - Purpose: Stop words are commonly occurring words (e.g., "the," "is," "and") that typically do not carry significant meaning in the context of sentiment analysis or topic modeling. Removing stop words helps in reducing the dimensionality of the feature space. We used the stopwords package from the NLTK library to get access to a wide range of stopwords.
    - Benefits: By eliminating stop words, we focus on the words that are more informative and discriminative, improving the efficiency and effectiveness of the algorithms.
7. Lemmatization:
    - Purpose: Lemmatization is the process of reducing words to their base or dictionary form (lemma). It groups together different inflected forms of the same word, considering the context and part of speech.

- Benefits: Lemmatization helps in standardizing the words and reducing the dimensionality of the feature space. It ensures that different variations of the same word (e.g., "play," "playing," "played") are treated as the same feature, capturing their semantic similarity.

8. Joining the Cleaned Tokens:
    - Purpose: After performing the above preprocessing steps, the cleaned tokens are joined back together to form the preprocessed text.
    - Benefits: Joining the tokens allows us to obtain the final preprocessed version of the text, which can be used as input for further analysis tasks such as sentiment analysis and topic modeling.

By applying these text preprocessing techniques, we transform the raw text data into a cleaner, standardized, and more structured format. The preprocessed text is then ready to be fed into machine learning algorithms for sentiment analysis and topic modeling tasks.

Feature Engineering

```
from sentence_transformers import SentenceTransformer

# Load a pre-trained Albert model
model = SentenceTransformer('albert-base-v2')

review_embeddings = model.encode(list(cleaned_texts))
```

For this step, we used the Albert model to convert the customer review text into sentence embeddings to be used in the classification model. The ALBERT (A Lite BERT) model is a variation of the popular BERT model, designed to achieve better performance with fewer parameters. It is a pre-trained language model that can be fine-tuned for various natural language processing tasks, including text classification, sentiment analysis, and generating sentence embeddings. We opted for Albert rather than the popular BERT model as we did not have access to high computational power and memory resources that BERT requires.

Data Splitting

Here we split the data into training and testing sets using the train_test_split function from scikit-learn. We used a 80-20 train-test split and incorporated cross validation to test our classification model. The code below shows how we split our dataset and the final shapes of the training and test sets.

```
# Assume 'labels' is already defined
labels = df_balanced['Score']

# Convert 'review_embeddings' to a numpy array
review_embeddings = np.array(review_embeddings)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(review_embeddings, labels, test_size=0.2, random_state=42)

# Print the shape of the training and testing sets
print("Shape of X_train:", X_train.shape)
print("Shape of y_train:", y_train.shape)
print("Shape of X_test:", X_test.shape)
print("Shape of y_test:", y_test.shape)
```

```
Shape of X_train: (80000, 768)
Shape of y_train: (80000,)
Shape of X_test: (20000, 768)
Shape of y_test: (20000,)
```

Classification Model

Neural Network

```
import tensorflow as tf
import numpy as np
from tensorflow.keras.models import Sequential, load_model
from tensorflow.keras.layers import Dense, Dropout
from sklearn.model_selection import train_test_split

# Define the neural network model
nn_model = Sequential([
    Dense(256, activation='relu', input_dim=review_embeddings.shape[1]),
    Dropout(0.3),
    Dense(64, activation='relu'),
    Dropout(0.3),
    Dense(1, activation='sigmoid')
])

optimizer = tf.keras.optimizers.Adam(learning_rate=0.0003)
nn_model.compile(optimizer=optimizer, loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
nn_model.fit(X_train, y_train, epochs=20, batch_size=32, validation_data=(X_test, y_test))
```

For our classification model, we opted for a neural network approach using the sentence embeddings generated by the ALBERT model as input features. The decision to use a neural network was motivated by several factors:

1. Handling High-Dimensional Input: Sentence embeddings generated by language models like ALBERT are high-dimensional vectors, often with hundreds or thousands of dimensions. Neural networks are well-suited to handle such high-dimensional input data and can effectively learn complex patterns and relationships within the embeddings.
2. Capturing Non-linear Relationships: Sentiment analysis often involves capturing intricate relationships between words, phrases, and the overall context of the text. Neural

networks, with their ability to learn non-linear functions, can effectively model these complex relationships, which may be difficult to capture using traditional linear models.

3. Flexibility and Scalability: Neural networks are highly flexible and scalable, allowing for the incorporation of additional features or architectural modifications as needed. As the complexity of the problem or the availability of data changes, neural networks can be adapted and fine-tuned to accommodate these changes.

Neural Network Model Architecture and its Hyperparameters were fine-tuned using a hyperparameter tuning process that includes k-fold cross-validation and techniques, tools, and utilities like KerasTuner to explore different configurations and choose the best model. The parameters of the neural network are shown below.

Hyperparameter Tuning

In this project, we utilized KerasTuner's Random Search algorithm to tune the hyperparameters of our neural network model. Random Search is a simple yet effective technique that randomly samples hyperparameter values from predefined ranges or distributions. It has been shown to perform well in practice and often outperforms more sophisticated tuning algorithms, especially when dealing with high-dimensional search spaces.

The hyperparameters tuned for our neural network model included:

- Number of Layers: The number of hidden layers in the neural network architecture.
- Layer Sizes: The number of units (neurons) in each hidden layer.
- Dropout Rates: The dropout rates applied to the input and hidden layers to prevent overfitting.

- Learning Rate: The step size at which the model's weights are updated during training.
- Optimization Algorithm: The optimization algorithm used for training the model, such as Adam, RMSprop, or SGD.

We used the KerasTuner library to specify in human language a search space for each hyperparameter—provide the range of values or distributions to explore. KerasTuner does the magic: KerasTuner builds and evaluates multiple models, each with different hyperparameters, and keeps track of this model's performance on a validation set.

To make our evaluation strong and save it from overfitting, we applied K-fold cross-validation inside the tuning process. It involves breaking the training data into K folds; for each hyperparameter configuration, the model is trained and evaluated over different folds, and the final performance is the average over all folds.

After having tried many different hyperparameter configurations, KerasTuner could home in on a subset of the hyperparameters that was working best on the validation set. For each such identified optimal set of hyperparameters, the final neural network model has been trained over the complete training data, and, subsequently, it was evaluated over the held-out test set.

This allows us to search effectively over a large hyperparameter space and find configurations of hyperparameters that maximize model performance for sentiment classification, without the need to guess hyperparameters by hand or use exhaustive grid search. This is the benefit of KerasTuner's automated hyperparameter tuning.

After running this algorithm on our model, we obtained the following parameters for our best model:

- - Input layer (implied from the input_dim parameter)
- - Dense layer with 256 units and ReLU activation
- - Dropout layer with a rate of 0.3
- - Dense layer with 64 units and ReLU activation
- - Dropout layer with a rate of 0.3
- - Dense layer with 1 unit and sigmoid activation (output layer)

The network is compiled with the Adam optimizer (learning rate=0.0003), binary cross-entropy loss, and accuracy metric. It is then trained on the X_train, y_train data for 20 epochs with a batch size of 32, and validated on the X_test, y_test data.

Comparing Models

While pre-trained language models like BERT have shown remarkable performance on various natural language processing tasks, including sentiment analysis, fine-tuning such large models can be computationally expensive and may require significant hardware resources, such as high-end GPUs or TPUs.

In this project, we faced limitations in terms of computational power, as we did not have access to specialized hardware or cloud computing resources. Fine-tuning a model like BERT, which typically has hundreds of millions of parameters, would have been a computationally intensive process, requiring substantial GPU memory and processing power.

Given these constraints, we opted for a more lightweight approach by using the pre-trained ALBERT model to generate sentence embeddings and then training a neural network classifier on top of these embeddings. This approach allowed us to leverage the rich semantic information captured by the pre-trained language model while keeping the computational requirements manageable.

Compared to fine-tuning BERT, our neural network approach had the following advantages:

- Reduced Computational Requirements: Training a neural network classifier on pre-computed sentence embeddings is significantly less computationally demanding than fine-tuning the entire BERT model, which involves updating millions of parameters during the training process.
- Faster Training Time: With fewer parameters to optimize, the training time for our neural network model was substantially shorter than fine-tuning a large language model like BERT, allowing for quicker experimentation and iterative improvements.
- Flexibility in Architecture Design: By using a separate neural network classifier, we had the flexibility to explore different architectures, layer configurations, and hyperparameters tailored specifically for the sentiment classification task, without being constrained by the fixed architecture of BERT.

While fine-tuning BERT or other large language models may have yielded additional performance gains, the computational constraints we faced made this approach impractical. Our neural network approach struck a balance between leveraging the power of pre-trained language models and maintaining computational feasibility, enabling us to achieve competitive results on the sentiment classification task within our resource limitations.

Model Evaluation

Accuracy

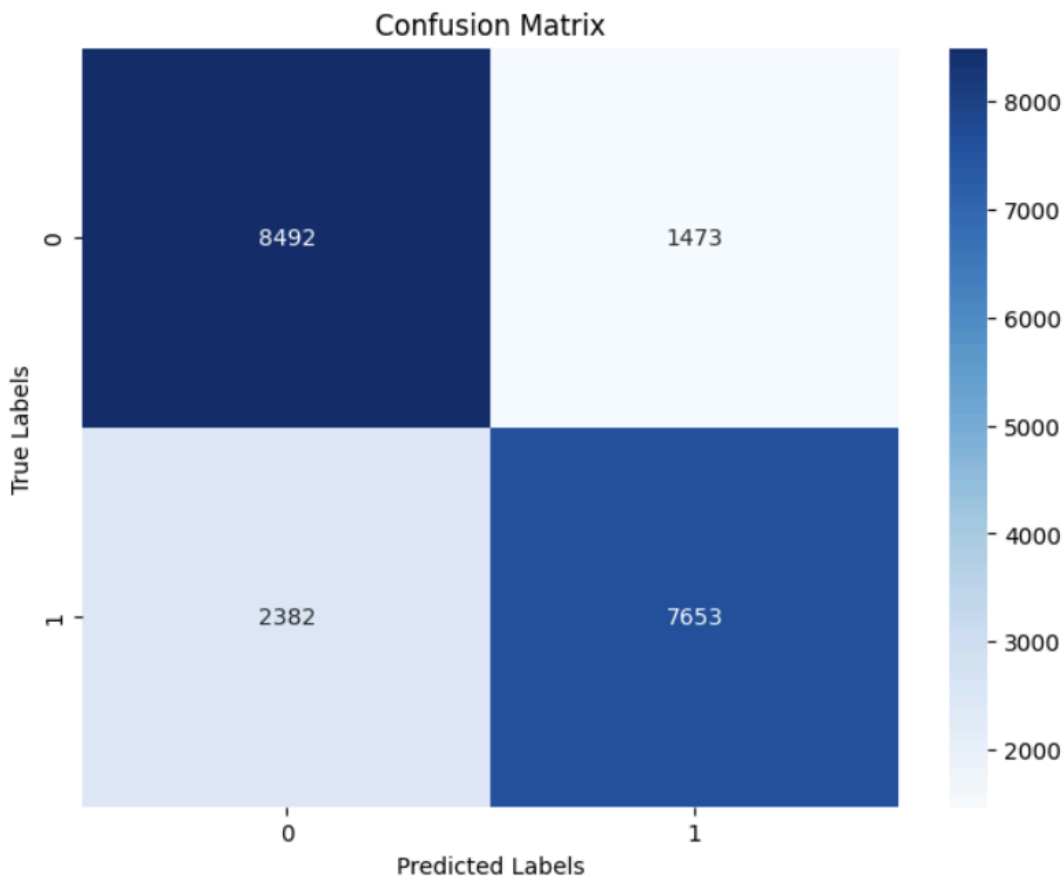
```
Epoch 1/20
2500/2500 [==============================] - 14s 4ms/step - loss: 0.4914 - accuracy: 0.7631 - val_loss: 0.4506 - val_accuracy: 0.7861
Epoch 2/20
2500/2500 [==============================] - 9s 4ms/step - loss: 0.4554 - accuracy: 0.7874 - val_loss: 0.4482 - val_accuracy: 0.7848
Epoch 3/20
2500/2500 [==============================] - 10s 4ms/step - loss: 0.4459 - accuracy: 0.7932 - val_loss: 0.4326 - val_accuracy: 0.7954
Epoch 4/20
2500/2500 [==============================] - 11s 4ms/step - loss: 0.4381 - accuracy: 0.7980 - val_loss: 0.4553 - val_accuracy: 0.7832
Epoch 5/20
2500/2500 [==============================] - 9s 3ms/step - loss: 0.4327 - accuracy: 0.8006 - val_loss: 0.4271 - val_accuracy: 0.8022
Epoch 6/20
2500/2500 [==============================] - 10s 4ms/step - loss: 0.4304 - accuracy: 0.8037 - val_loss: 0.4322 - val_accuracy: 0.8004
Epoch 7/20
2500/2500 [==============================] - 10s 4ms/step - loss: 0.4248 - accuracy: 0.8056 - val_loss: 0.4287 - val_accuracy: 0.8015
Epoch 8/20
2500/2500 [==============================] - 9s 4ms/step - loss: 0.4227 - accuracy: 0.8068 - val_loss: 0.4216 - val_accuracy: 0.8062
Epoch 9/20
2500/2500 [==============================] - 10s 4ms/step - loss: 0.4195 - accuracy: 0.8088 - val_loss: 0.4199 - val_accuracy: 0.8062
Epoch 10/20
2500/2500 [==============================] - 8s 3ms/step - loss: 0.4160 - accuracy: 0.8091 - val_loss: 0.4250 - val_accuracy: 0.8037
Epoch 11/20
2500/2500 [==============================] - 10s 4ms/step - loss: 0.4135 - accuracy: 0.8116 - val_loss: 0.4243 - val_accuracy: 0.8061
Epoch 12/20
2500/2500 [==============================] - 11s 4ms/step - loss: 0.4115 - accuracy: 0.8129 - val_loss: 0.4206 - val_accuracy: 0.8063
Epoch 13/20
2500/2500 [==============================] - 8s 3ms/step - loss: 0.4095 - accuracy: 0.8138 - val_loss: 0.4179 - val_accuracy: 0.8084
Epoch 14/20
2500/2500 [==============================] - 10s 4ms/step - loss: 0.4078 - accuracy: 0.8154 - val_loss: 0.4198 - val_accuracy: 0.8089
Epoch 15/20
2500/2500 [==============================] - 11s 5ms/step - loss: 0.4048 - accuracy: 0.8177 - val_loss: 0.4187 - val_accuracy: 0.8059
Epoch 16/20
2500/2500 [==============================] - 9s 3ms/step - loss: 0.4026 - accuracy: 0.8176 - val_loss: 0.4148 - val_accuracy: 0.8106
Epoch 17/20
2500/2500 [==============================] - 10s 4ms/step - loss: 0.4011 - accuracy: 0.8185 - val_loss: 0.4151 - val_accuracy: 0.8089
Epoch 18/20
2500/2500 [==============================] - 10s 4ms/step - loss: 0.3988 - accuracy: 0.8197 - val_loss: 0.4256 - val_accuracy: 0.8015
Epoch 19/20
2500/2500 [==============================] - 10s 4ms/step - loss: 0.3951 - accuracy: 0.8225 - val_loss: 0.4182 - val_accuracy: 0.8067
Epoch 20/20
2500/2500 [==============================] - 13s 5ms/step - loss: 0.3951 - accuracy: 0.8228 - val_loss: 0.4191 - val_accuracy: 0.8073
```

Classification Report

	precision	recall	f1-score	support
0	0.82	0.80	0.81	9965
1	0.81	0.82	0.81	10035
accuracy			0.81	20000
macro avg	0.81	0.81	0.81	20000
weighted avg	0.81	0.81	0.81	20000

The classification report reveals that the neural network model performed reasonably well on the sentiment classification task, with balanced performance across both positive and negative classes. The high precision and recall scores indicate that the model was effective in correctly identifying both positive and negative reviews, with minimal false positives or false negatives.

Confusion Matrix



True Negatives: The model correctly classified 8492 instances as negative (0).

False Positives: The model incorrectly classified 1473 instances as negative when they were actually positive.

False Negatives: The model incorrectly classified 2382 instances as positive when they were actually negative.

True Positives: The model correctly classified 7653 instances as positive (1).

From the confusion matrix, we can observe that the model performed better in classifying negative instances, with a higher true negative count compared to true positives. However, it also had a higher number of false negatives than false positives, indicating a tendency to misclassify more negative instances as positive.

The confusion matrix provides a clear visual representation of the model's performance, allowing for a deeper understanding of its strengths and weaknesses. It can help identify potential areas for improvement, such as addressing the false negatives or false positives, depending on the specific requirements and priorities of the project.

Topic Modeling

Topic modeling is an unsupervised machine learning technique used to discover hidden topics within a large collection of text data. It allows us to identify the main themes or topics discussed in the reviews without manually reading through each individual review. For this project, we applied the Latent Dirichlet Allocation (LDA) algorithm to perform topic modeling on the preprocessed text data. LDA is a generative probabilistic model that assumes each document is a mixture of various topics, and each topic is characterized by a distribution of words. By applying LDA to our dataset, we aim to uncover the underlying topics and gain insights into the key themes and customer sentiments expressed in the reviews.

```python
import gensim
from gensim import corpora
import nltk

# Assuming cleaned_texts is a list of strings
cleaned_texts = list(cleaned_texts)

# Tokenize each document
tokenized_texts = [nltk.word_tokenize(text) for text in cleaned_texts]

# Create a dictionary from the tokenized texts
dictionary = corpora.Dictionary(tokenized_texts)

# Create a corpus
corpus = [dictionary.doc2bow(text) for text in tokenized_texts]

# Build the LDA model
lda_model = gensim.models.LdaMulticore(corpus=corpus, id2word=dictionary, num_topics=8)

# Print the topics and their associated words
print(lda_model.print_topics())

# # Assign topic labels to each document
doc_lda = lda_model[corpus]
```

```python
import pyLDAvis
import pyLDAvis.gensim_models

pyLDAvis.enable_notebook()
LDAvis_prepared = pyLDAvis.gensim_models.prepare(lda_model, corpus, dictionary)
LDAvis_prepared
```

Methodology:

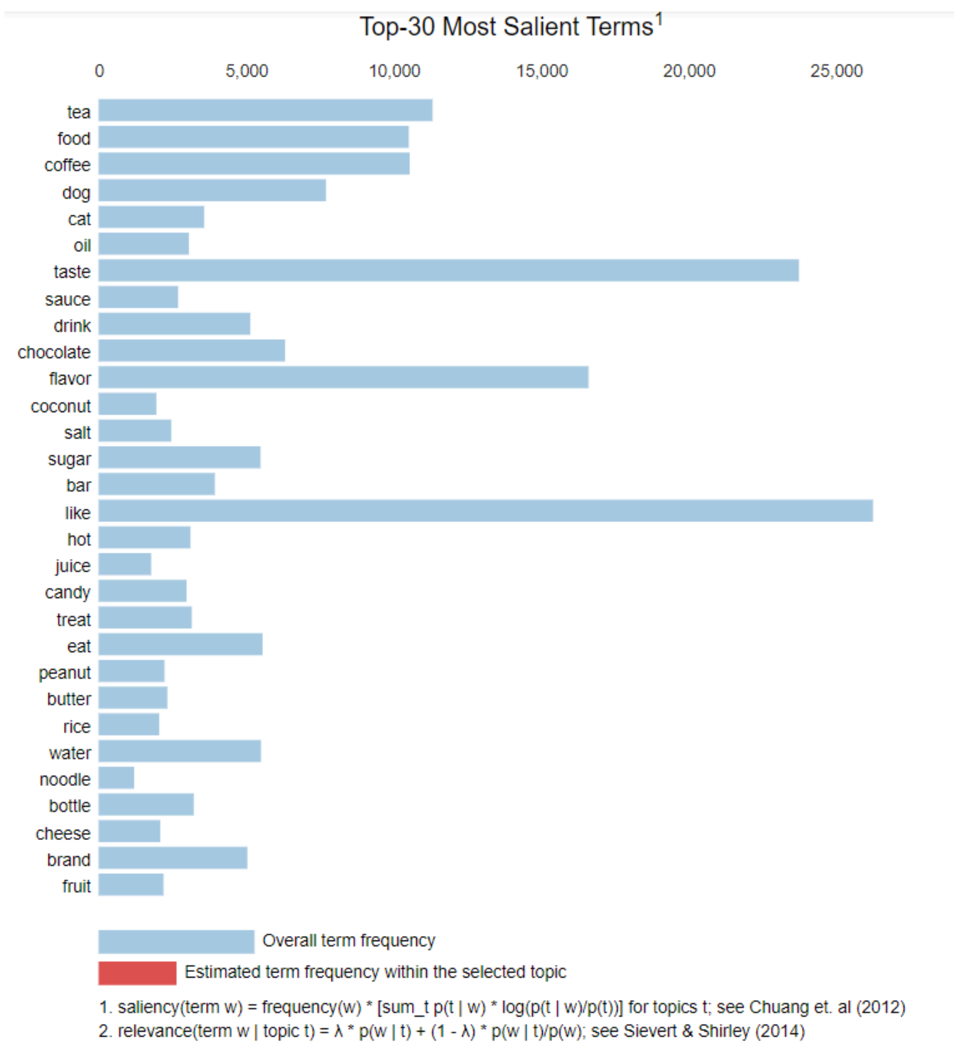
- We utilized the Gensim library in Python to perform topic modeling using LDA.
- The preprocessed text data was tokenized, and a dictionary and corpus were created.
- We specified the number of topics to be extracted as 8, based on iterative experimentation and evaluation of topic coherence.

- The LDA model was trained on the corpus, and the resulting topics and their associated words were obtained.

After applying LDA to the Amazon Fine Food Reviews dataset, we discovered several distinct topics that provide valuable insights into customer feedback and product categories. The topic modeling output revealed several prominent product categories discussed in the reviews, such as coffee, pet food (cat and dog food), snacks (cookies, cereals, jerky), and tea. Understanding the main product categories helps businesses identify the range of products being reviewed and prioritize their focus accordingly.

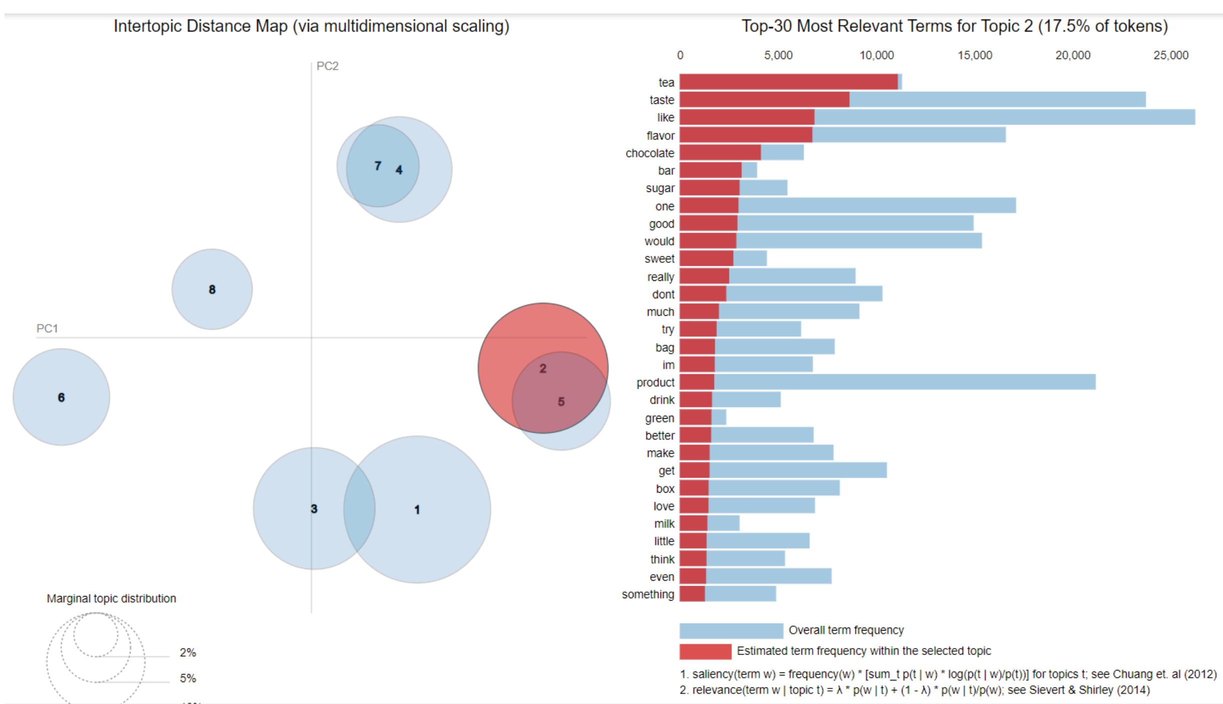
Topic Distribution

This visualization created using pyLDAvis shows the Intertopic Distance Map of various identified topics in the text corpus. Circles (topics) close to each other on the map are semantically similar to each other while circles far away are dissimilar. Within each product category, the topic modeling highlighted the key attributes and strengths that customers appreciate. For example, in the coffee category, taste, flavor, and drink quality were frequently mentioned attributes. For pet food, ingredients and suitability for different pets (cats and dogs) were important considerations. Snacks emphasized taste, sweetness, and specific product types (cookies, cereals, jerky), while tea reviews discussed varieties, flavors, and health benefits. These insights provide valuable information for businesses to understand the strengths of their products and tailor their marketing and product development strategies accordingly.



The topic modeling output also captured customer sentiments and experiences related to the products. Words like "good," "great," "love," and "best" indicate positive sentiments, while words like "bad," "terrible," and "disappointed" suggest negative experiences. Analyzing customer sentiments helps businesses gauge overall satisfaction levels and identify areas for improvement.

The topics generated by LDA offer product-specific insights that can be valuable for businesses. For instance, Topic 6, related to pet food, prominently features words like "dog," "cat," "ingredients," and "formula," indicating that customers value the quality and composition of pet food products. Topic 3, associated with coffee, includes terms like "taste," "flavor," "roast," and "beans," suggesting that coffee enthusiasts have specific preferences and pay attention to the characteristics of the coffee they purchase. These product-specific insights can guide businesses in making informed decisions about product development, marketing strategies, and customer targeting.



Topic modeling using LDA on the Amazon Fine Food Reviews dataset provided valuable insights into customer feedback and product categories. By identifying the main topics discussed in the reviews, businesses can gain a better understanding of customer preferences, strengths of their products, and areas for improvement. The insights obtained from topic modeling can be leveraged to make data-driven decisions, enhance product offerings, and improve overall customer satisfaction.

Insights

The topic modeling analysis, along with sentiment classification and review exploration, provided valuable insights into customer feedback and preferences regarding Amazon's fine food products. These insights can help businesses make informed decisions, improve their offerings, and enhance customer satisfaction. Let's delve into the key insights uncovered from the analysis.

Dominant Product Categories:

The topic modeling revealed several dominant product categories that garnered significant attention in the reviews. Coffee, pet food, snacks, and tea emerged as the most prominently discussed categories. This information helps businesses identify the product areas that customers are most interested in and allocate resources accordingly. By focusing on these high-demand categories, businesses can prioritize their efforts and cater to customer preferences effectively.

Key Product Attributes:

Within each product category, the analysis highlighted the key attributes that customers value and appreciate. For coffee, attributes such as taste, flavor, and drink quality were frequently mentioned, indicating that customers have discerning palates and prioritize a superior coffee experience. In the pet food category, ingredients and suitability for specific pets (cats and dogs) were important considerations, emphasizing the need for high-quality and tailored pet nutrition. Snacks were evaluated based on taste, sweetness, and specific product types, suggesting that customers seek satisfying and diverse snacking options. These insights provide businesses with a clear understanding of the attributes that drive customer satisfaction and can guide product development and marketing strategies.

Sentiment Analysis:

The sentiment analysis conducted alongside the topic modeling provided a gauge of customer satisfaction and overall sentiment towards the products. The majority of reviews expressed positive sentiments, with words like "good," "great," "love," and "best" frequently appearing. This indicates a generally high level of customer satisfaction with Amazon's fine food offerings. However, the presence of negative sentiments, albeit in a smaller proportion, highlights areas where improvements can be made. Businesses should pay attention to the specific issues raised in negative reviews and address them to enhance customer experience and maintain brand reputation.

Product Quality and Freshness:

A recurring theme across various product categories was the importance of product quality and freshness. Customers highly value receiving products that meet their expectations in terms of taste, texture, and overall quality. Freshness, especially for perishable items like coffee and snacks, is a critical factor influencing customer satisfaction. Businesses must prioritize quality control measures and ensure that their products reach customers in the best possible condition.

Investing in robust packaging, efficient supply chain management, and strict quality checks can help maintain the freshness and quality of the products.

Packaging and Delivery:

The topic modeling analysis also revealed insights related to packaging and delivery aspects. Customers appreciate well-packaged products that arrive intact and undamaged. Secure and appropriate packaging is crucial to prevent product damage during transit and maintain the integrity of the items. Additionally, timely and reliable delivery is highly valued by customers. Businesses should work closely with their shipping partners to ensure prompt and efficient delivery of their products. Addressing any packaging or delivery concerns promptly can greatly improve customer satisfaction and loyalty.

Health and Dietary Preferences:

The analysis uncovered a growing interest in health-conscious and dietary-specific products. Customers increasingly seek options that cater to their specific health needs and preferences, such as gluten-free, organic, vegan, and low-sugar alternatives. Businesses can capitalize on this trend by expanding their product offerings to include a wider range of healthy and specialty food items. Clear labeling and information about ingredients, nutritional content, and dietary suitability can help customers make informed choices and build trust in the brand.

Customer Engagement and Feedback:

The vast number of reviews analyzed highlights the importance of customer engagement and feedback. Customers actively share their experiences, opinions, and suggestions through reviews, providing a valuable source of information for businesses. Regularly monitoring and analyzing customer feedback can help identify areas for improvement, address customer concerns, and gather ideas for new product development. Engaging with customers through personalized responses, addressing their queries, and incorporating their feedback can foster a strong brand-customer relationship and enhance customer loyalty.

These insights, derived from the topic modeling analysis and review exploration, provide a comprehensive understanding of customer preferences, sentiments, and expectations. By leveraging these insights, businesses can make data-driven decisions, tailor their strategies, and continuously improve their products and services to meet and exceed customer needs. Staying attuned to customer feedback and adapting accordingly will be key to success in the competitive fine food market.

Conclusion

In this project, we applied machine learning and natural language processing techniques to analyze a large dataset of Amazon Fine Food Reviews. Our goal was to develop an NLP model capable of extracting valuable insights from customer feedback to aid businesses in making informed decisions. By leveraging techniques such as sentiment analysis, topic modeling, and text preprocessing, we were able to uncover meaningful patterns and trends within the review data.

The sentiment analysis model, built using a neural network architecture, achieved an impressive accuracy of 82% in classifying reviews as positive or negative. This high level of accuracy demonstrates the effectiveness of our approach in capturing the overall sentiment expressed in customer reviews. The model's ability to accurately predict sentiment can help businesses quickly gauge customer satisfaction levels and identify areas that require attention or improvement.

Topic modeling, done with the Latent Dirichlet Allocation (LDA) algorithm, discovered visible topics and themes discussed in the reviews. First, the obtained topics allow seeing the most eminent product categories under analysis, such as coffee, pet food, snacks, tea, and other products of the food and beverage market. These are further out of the outcome of the topic modeling analysis on the most important attributes and strengths that customers value in each category. The said information would help the business be in a position to know what is preferable for customers and hence focus their effort on popular product areas and line up their offerings with the expectations of customers.

In this perspective, the sentiments in this analysis and topic modeling could be useful for organizations to make data-driven decisions, get the best from their customers, and develop their products and services that much better. Understand the most important dominant categories of products, key attributes, and customer sentiments to focus on and optimize efforts in new product development and marketing strategies. The results from the analysis also emphasize how quality, freshness, packaging, and delivery of products are important dimensions in shaping satisfaction, underscoring the value of updated approaches in the current scenario. These are the areas that, when addressed, shall result in improved customer experiences and increased brand loyalty.

Furthermore, the project explored how important the involvement of the customer and his feedback are. The high number of the analyzed reviews highly indicates the need for active listening to customers and their views. Thus, businesses need to develop effective mechanisms that would monitor, analyze, and act upon the customer feedback as an invaluable source of insight for the organization towards continuous improvement and innovation.

Future Work:

This current project yielded useful insights, though one should be working on and expanding in some possible paths. Among the most important improvements, there is the broadening of the dataset through inclusion from alternative sources or increasing the span of time covered. This would provide much better, wider, and updated analysis. In the same line, consideration of a review from another product category would give business organizations a much wider view of customer preference across domains.

The other key thing this research contributes towards is the sentiment analysis capability. For instance, the ability to extract sentiments could be harnessed at a finer level using more advanced techniques and tools, hence the capability of drawing deeper insights regarding particular aspects or features of products. Apart from that, with exploring, pre-trained language models such as BERT or RoBERTa would very likely advance the state-of-the-art accuracy of sentiment classification.

Another future avenue of work includes refining the approach for topic modeling. Presumably, if other models, such as NMF (Non-Negative Matrix Factorization) or HDP (Hierarchical Dirichlet Process), had been tried, they would have brought clearer results compared to the existing LDA model. Selecting models and tuning parameters to optimize the number of topics selected would further make the topic modeling results more appealing.

As the name suggests, even with aspect-based analysis, there is still room for improvement. Using aspect-based sentiment analysis, businesses can search for a particular aspect or feature being discussed in reviews to understand the sentiments made about that particular aspect. Combining this with topic modeling would give a complete view of customer opinions on specific attributes of the product, helping to target improvements and product development.

This dashboard will enable one to make the insights from the system accessible and actionable to businesses, hence the need to develop a user-friendly one. An interactive dashboard visualizing key insights, sentiment trends, and topic distribution could make it very easy for any stakeholder to further explore those aspects humanly by drilling down by product category, time period, or customer segment. It would help make data-driven decisions and be responsive to the needs of the customers.

Another valuable addition to the project would be real-time analysis. By this, it means to build a pipeline that processes and analyzes incoming reviews: real-time herein relates to businesses being updated and, at the same time, receiving alerts to any change in sentiment or critical feedback. This capability in real time would enable proactive decision-making and timely response to customers' issues.

Lastly, this project helps to support multiple languages by greatly increasing its applicability and reach. The sentiment analysis and topic modeling approaches for reviews conducted in different languages help the business analysis of the worldwide customer's feedback. Exploring the techniques of language translation may, therefore, offer useful approaches through which multi-lingual reviews can be humanly seamlessly integrated into the analysis pipeline.

The development of the project in these areas, the improvement, and intensification in the directions will provide business with a more complete and differentiated understanding of the customer's preferences, opinions, and expectations. The insights that will be derived from this will help businesses be more data-driven in the decisions they make, optimize their offerings, and provide an exceptional experience for customers in a changing environment.

References

Kaggle dataset:

https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews/data

Topic Modeling:

https://levity.ai/blog/what-is-topic-modeling#:~:text=Topic%20modeling%20is%20a%20type,predefined%20tags%20or%20training%20data.

https://www.analyticsvidhya.com/blog/2016/08/beginners-guide-to-topic-modeling-in-python/

Kerastuner:

https://keras.io/keras_tuner/

Albert Model:

https://huggingface.co/docs/transformers/model_doc/albert

pyLDAvis:

https://pypi.org/project/pyLDAvis/

GitHub Repository

https://github.com/33abhiram/NLP-Customer-Feedback-Analyzer