Technical Report: Sentiment Analysis and Topic Modeling on Amazon Fine Food Reviews

Project Overview and Objectives

This project involves performing sentiment analysis and topic modeling on the Amazon Fine Food Reviews dataset. The primary objective is to understand customer sentiments based on their reviews and extract key topics discussed in both positive and negative reviews. This involves:

- **Sentiment Analysis:** Classifying reviews as positive or negative based on the rating score.
- **Topic Modeling:** Identifying key topics from the review text using Latent Dirichlet Allocation (LDA).
- **Data Visualization:** Creating visual representations of the sentiment distribution over time, word clouds of the topics, and correlation between identified topics and sentiment.

Methodology

The methodology was divided into several phases:

1. Data Preparation and Exploratory Data Analysis (EDA)

Data Loading and Initial Exploration:

- The dataset was loaded using `pandas`, containing information such as review text, review rating (Score), and the time of the review.
- The dataset includes over 500,000 customer reviews, and the first step involved converting the `Time` column from Unix timestamps to a human-readable format using `pandas.to_datetime()`.

Exploratory Data Analysis (EDA):

- **Sentiment Distribution:** Sentiment was categorized as *Positive* (Score ≥ 4) and *Negative* (Score < 4). A sentiment distribution plot was generated to analyze how customer sentiment changed over time using `seaborn`.
- **Common Words:** The most frequent words in positive and negative reviews were identified. We visualized the top 10 words for both sentiments using bar charts.

2. Text Preprocessing and Feature Extraction

Text Cleaning and Tokenization:

Reviews were preprocessed by converting them to lowercase, tokenizing using `nltk.word_tokenize()`, and removing stop words from the `nltk.corpus.stopwords` list. The resulting tokens were stored in a new `Tokens` column.

python

Copy code

```
def preprocess_text(text):
    tokens = word_tokenize(text.lower())
    tokens = [word for word in tokens if word.isalpha() and word not in stop_words]
    return tokens
```

Feature Extraction:

- The preprocessed reviews were tokenized, and Term Frequency-Inverse Document Frequency (TF-IDF) vectorization was applied using `scikit-learn` to transform text data into numerical form.

3. Sentiment Analysis

Modeling with Machine Learning:

Logistic Regression was chosen for the binary sentiment classification task (positive vs. negative reviews). We used features such as `HelpfulnessNumerator`, `HelpfulnessDenominator`, and the review `Score`.

```
X = data[['HelpfulnessNumerator', 'HelpfulnessDenominator', 'Score']]
y = np.where(data['Score'] >= 4, 1, 0)  # Simplified sentiment
```

-
- The model was trained using `train_test_split` for a train-test ratio of 80:20, followed by evaluating using accuracy, precision, recall, and F1-score.

4. Topic Modeling with Latent Dirichlet Allocation (LDA)

Preprocessing for LDA:

- Stop words were removed, and tokens were created from the text. The `gensim.Dictionary` was used to create a dictionary mapping each unique word to an ID. A Bag of Words (BoW) corpus was constructed for LDA input.

LDA Training:

LDA with 5 topics was trained using `gensim.models.LdaModel`. The number of topics was optimized based on coherence scores, and the model was trained with 15 passes over the corpus.

```
lda_model = LdaModel(corpus, num_topics=5, id2word=dictionary,
passes=15)
```

Topic Visualization:

- **Word Clouds** were generated for each topic, providing an intuitive visualization of the key words that define each topic.

Topic-Sentiment Correlation:

- We extracted topic distributions for each review and correlated them with the sentiment labels to explore how strongly each topic was associated with positive or negative sentiment. The correlation was visualized using a heatmap.

5. Data Visualization

Sentiment Distribution Over Time:

- Using `seaborn`, a count plot was created to visualize how the sentiment distribution has evolved over the years.

Word Clouds:

- For each identified topic, word clouds were generated using the `WordCloud` library, providing a visual representation of the most frequent words associated with that topic.

Bar Charts of Top Words:

- Bar charts showed the most important words for both positive and negative sentiment reviews. For example, in positive reviews, words like "love," "great," and "good" dominated, while negative reviews contained words like "bad," "disappointed," and "poor."

Heatmap of Topic-Sentiment Correlation:

- The correlation between each topic and the sentiment was calculated and visualized using a heatmap, showing which topics were most strongly associated with positive or negative sentiment.

Results and Interpretation

- **Sentiment Distribution:** The distribution of sentiment over time shows a relatively consistent pattern, with positive reviews significantly outnumbering negative reviews throughout the dataset.
- **Topic Modeling:** Five distinct topics were identified, and word clouds provided insight into the nature of these topics. For example, one topic was centered around product quality, while another focused on customer service.
- **Correlation Analysis:** Topics correlated with positive sentiment were typically associated with high product satisfaction (e.g., "delicious," "love"), while topics associated with negative sentiment highlighted issues with packaging or delivery.

Challenges Faced and Solutions Implemented:

Convergence Warning in Logistic Regression: The Logistic Regression model encountered convergence issues, as the lbfgs solver failed to converge. This was addressed by adjusting the max_iter parameter and considering alternative solvers as suggested by the scikit-learn documentation.

SVM and Random Forest Training Times: The training time for SVM and Random Forest models was prohibitively long. As a result, these models were not used in the final evaluation, and faster models like Naive Bayes and LightGBM were selected.

Class Imbalance: The dataset was highly imbalanced, with significantly more positive sentiment samples than negative ones. This was addressed by using appropriate performance metrics (precision, recall, and f1-score) in addition to accuracy.

Training Time for Deep Learning Models: The LSTM model required substantial computational resources and training time. This was mitigated by using a batch size of 64 and early stopping to avoid overfitting.

Future Improvements and Potential Applications

- **Advanced Models:** Future work could involve experimenting with more sophisticated models like BERT for sentiment analysis to capture the context of reviews more effectively.
- **Sentiment and Topic Insights:** These insights can help businesses improve their products and services by focusing on customer feedback trends over time.
- **Real-time Analysis:** Implementing real-time topic modeling and sentiment analysis on customer feedback could provide valuable, actionable insights as new reviews are submitted.