Sentiment Analysis of Jumia Fashion Product Reviews

1. Introduction

1.1 Objective: The goal of this project is to conduct sentiment analysis on fashion product reviews from the Nigerian e-commerce platform Jumia, using Bag-of-Words (BoW) features and various machine learning models - Naive Bayes, Support Vector Machines (SVM), Logistic Regression and XGBoost. The analysis aims to categorize the sentiment of the reviews into five classes: Positive, Slightly Positive, Neutral, Slightly Negative, and Negative. This analysis provides valuable insights into customer satisfaction and areas for product improvement, which are critical for businesses in the rapidly growing Nigerian e-commerce market.

1.2 Background: In the competitive e-commerce landscape, understanding customer feedback is essential for improving products and services. Sentiment analysis helps businesses capture the underlying sentiment in customer reviews, offering a structured approach to enhancing customer satisfaction and product offerings. This project is set within the context of the Nigerian market, focusing on fashion product reviews from Jumia, one of the leading e-commerce platforms in Nigeria.

1.3 Problem Statement: The problem this project aims to solve is to automatically categorize customer reviews into different sentiment classes to better understand customer feedback and improve product offerings. In the fashion industry, where customer satisfaction is key, this analysis can help businesses tailor their products and services to meet customer needs more effectively.

1.4 Significance: The insights gained from this sentiment analysis are crucial for businesses to understand customer sentiment and make data-driven decisions. By identifying common themes in positive and negative reviews, businesses can enhance product quality, improve customer satisfaction, and ultimately drive sales growth.

---

2. Data Collection and Labeling

2.1 Data Collection: For this project, product reviews were collected from jumia.com.ng (Fashion Section). The dataset comprises 3024 reviews, which were gathered and prepared for analysis. Each review includes metadata such as product information and customer ratings.

```
     id               Name        Date          Title  \
0  6414              Memee  09/08/2024  lovely quality
1  6415             Folake  09/08/2024           good
2  6416  Abayomi Olowosoyo  09/08/2024      Very good
3  6417            oladipo  05/08/2024       Thumb up
4  6418             Sharon  16/07/2024        Like it

                        text  Rating  \
0  Came as seen. Good product!       5
1    It's a very nice material       4
2                         Nice       5
3       portable & comfortable       4
4                    Beautiful       5

                                         Product_url              label
0  https://www.jumia.com.ng/vejaro-bc003-infant-b...           Positive
1  https://www.jumia.com.ng/vejaro-bc003-infant-b...  Slightly Negative
2  https://www.jumia.com.ng/vejaro-bc003-infant-b...           Positive
3  https://www.jumia.com.ng/fashion-womens-diamon...  Slightly Negative
4  https://www.jumia.com.ng/fashion-womens-diamon...           Positive
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3024 entries, 0 to 3023
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   id           3024 non-null   int64
 1   Name         3024 non-null   object
 2   Date         3024 non-null   object
 3   Title        3024 non-null   object
 4   text         3024 non-null   object
 5   Rating       3024 non-null   int64
 6   Product_url  3024 non-null   object
 7   label        3024 non-null   object
dtypes: int64(2), object(6)
memory usage: 189.1+ KB
None
id             0
Name           0
Date           0
Title          0
```

- **Source:** Jumia.com.ng (Fashion Section)
- **Content:** Product reviews collected from the fashion section.
- **Labeling Tool:** Doccano was employed to manually label the sentiment of each review.
- **Dataset Size:** The dataset consists of 3,024 entries, each containing a review, a sentiment label and additional metadata such as rating and product URL.

2.2 Approach to Data Collection: The data collection process involved the following steps:

- **Web Scraping Setup**:
  - **Libraries Used**: I used Python libraries such as requests for making HTTP requests, BeautifulSoup for parsing HTML content, and pandas for data manipulation and storage.
  - **Retry Logic**: To ensure the reliability of the scraping process, especially given the potential for network issues, a session with retry logic was implemented. This session automatically retries failed requests, reducing the likelihood of data loss or interruptions during scraping.
- **Product URL Collection**:
  - The first step in the scraping process was to collect URLs of individual product pages. I focused on the fashion category, iterating through multiple listing pages on Jumia.
  - **Checkpointing**: To avoid reprocessing and ensure that progress was saved in case of interruptions, the product URLs were periodically saved to a CSV file.
- **Review Scraping**:
  - For each product URL collected, the script navigated to the corresponding page and scraped reviews. The reviews included details such as the reviewer's name, rating, review title, review body, and the date of the review.
  - **Pagination Handling**: Many product pages have multiple pages of reviews. The script was designed to handle pagination, ensuring that all reviews for each product were collected as well.

- **Data Storage**:
  - The scraped reviews were stored in a structured format within a Pandas DataFrame and periodically saved to a CSV file. This not only ensured data persistence but also made it easier to process the data in subsequent tasks.

2.3 Data Labeling with Doccano: The reviews were labeled using Doccano, a text annotation tool.

	ID	Status	Text	Metadata	Comments	Action
☐	6414	Finished	Came as seen. Good product!	{ "Name": "Memee", "Rating": "5", "Date": "09/08/2024", "Title": "lovely quality", "Product_url": "https://www.jumia.com.ng/vejaro-bc003-infant-baby-clothes-sets-dinosaur-yellow-stripe-242036916.html" }	0	Edit Annotate
☐	6415	Finished	It's a very nice material	{ "Name": "Folake", "Rating": "4", "Date": "09/08/2024", "Title": "good", "Product_url": "https://www.jumia.com.ng/vejaro-bc003-infant-baby-clothes-sets-dinosaur-yellow-stripe-242036916.html" }	0	Edit Annotate
☐	6416	Finished	Nice	{ "Name": "Abayomi Olowosoyo", "Rating": "5", "Date": "09/08/2024", "Title": "Very good", "Product_url": "https://www.jumia.com.ng/vejaro-bc003-infant-baby-clothes-sets-dinosaur-yellow-stripe-242036916.html" }	0	Edit Annotate
☐	6417	Finished	portable & comfortable	{ "Name": "oladipo", "Rating": "4", "Date": "05/08/2024", "Title": "Thumb up", "Product_url": "https://www.jumia.com.ng/fashion-womens-diamond-rectangular-gold-wristwatch-234192668.html" }	0	Edit Annotate
☐	6418	Finished	Beautiful	{ "Name": "Sharon", "Rating": "5", "Date": "16/07/2024", "Title": "Like it", "Product_url": "https://www.jumia.com.ng/fashion-womens-diamond-rectangular-gold-wristwatch-234192668.html" }	0	Edit Annotate
☐	6419	Finished	Perfectly matched	{ "Name": "Favour", "Rating": "3", "Date": "09/07/2024", "Title": "gold and sweet", "Product_url": "https://www.jumia.com.ng/fashion-womens-diamond-rectangular-gold-wristwatch-234192668.html" }	0	Edit Annotate
☐	6420	Finished	Recommended	{ "Name": "Ayodele", "Rating": "5", "Date": "07/08/2024", "Title": "Smart", "Product_url": "https://www.jumia.com.ng/fashion-canvas-shoes-trendy-anti-slip-breathable-sneaker-for-men-flats-black-249148412.html" }	0	Edit Annotate
☐	6421	Finished	Simple and smart	{ "Name": "Iheanacho", "Rating": "5", "Date": "31/07/2024", "Title": "sharp in my legs", "Product_url": "https://www.jumia.com.ng/fashion-canvas-shoes-trendy-anti-slip-breathable-sneaker-for-men-flats-black-249148412.html" }	0	Edit Annotate

The reviews were categorized into five sentiment classes:

- **Positive (5)**
- **Slightly Positive (4)**
- **Neutral (3)**
- **Slightly Negative (2)**
- **Negative (1)**

Search

	Name	Shortkey	Color	Actions
☐	Positive	5	#68BC00	
☐	Slightly Positive	4	#A4DD00	
☐	Neutral	3	#73D8FF	
☐	Slightly Negative	2	#F44E3B	
☐	Negative	1	#D33115	

Rows per Page 10 1-5 of 5

These classes were chosen to capture a granular understanding of customer sentiment, ranging from highly positive to highly negative.

2.4 Challenges in Data Collection and Labeling:

- **Bias and Fairness:** There may be potential biases in the data, such as overrepresentation of certain types of products or customer demographics. This could affect the generalizability of the model.
- **Computational Resources:** The web scraping process, particularly handling pagination and large volumes of data, required significant computational resources. To mitigate this, the scraping was performed incrementally with checkpointing to ensure no data was lost in case of interruptions especially from network issues.

---

3. Data Preprocessing

3.1 Text Cleaning: The textual data underwent a series of preprocessing steps to prepare it for model training:

- **Lowercasing**: All text was converted to lowercase to ensure uniformity.
- **Removing Stop Words**: Common words that do not contribute significantly to the sentiment were removed.

3.2 Tokenization: Tokenization was performed to split the text into individual words or tokens. This process is essential for feature extraction using the Bag-of-Words model.

3.3 Handling Class Imbalance: The dataset was heavily imbalanced, with a predominant number of positive reviews. This imbalance could have led to biased model predictions, where the model overly predicts the majority class. To address the issue of class imbalance, the minority classes were upsampled. This ensured that each sentiment category had an equal number of samples (1367), leading to a more balanced dataset. The upsampling process improved model training by providing equal representation across all sentiment classes. These visualizations demonstrate the initial class imbalance, which was addressed by upsampling the minority classes to create a more balanced dataset for model training.

HANDLING CLASS IMBALANCE

In [327]:
```python
# Upsample Minority size
majority_class_size = df['label'].value_counts().max()

# Create an empty list to store the upsampled DataFrames
upsampled_dfs = []

# Loop through each class and upsample if it's smaller than the majority class size
for label in df['label'].unique():
    df_class = df[df['label'] == label]
    if len(df_class) < majority_class_size:
        df_class_upsampled = resample(df_class,
                                      replace=True,
                                      n_samples=majority_class_size,
                                      random_state=42)
    else:
        df_class_upsampled = df_class
    upsampled_dfs.append(df_class_upsampled)

# Combine all the upsampled classes into one DataFrame
df_upsampled = pd.concat(upsampled_dfs)

# Check the new class distribution
print(df_upsampled['label'].value_counts())
```

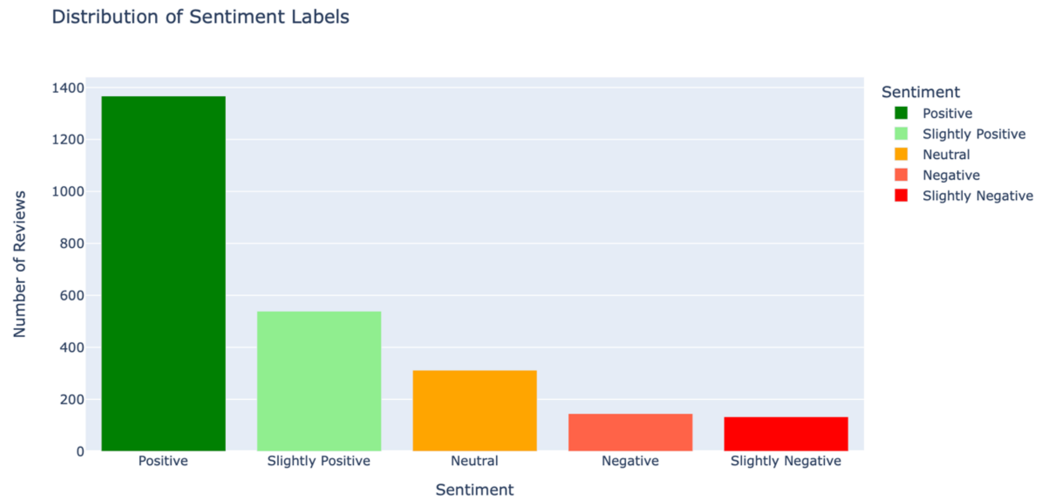
```
Positive             1367
Slightly Negative    1367
Neutral              1367
Slightly Positive    1367
Negative             1367
Name: label, dtype: int64
```

3.4 Other Challenges in Data Preprocessing:

- **Noisy Data:** Handling noisy text, including special characters and varying formats, was a challenge. Regular expressions were employed to clean the text effectively.
- **Handling Special Characters:** Deciding whether to keep or remove special characters, emoticons, or emojis was critical in maintaining the sentiment's integrity.

4 Exploratory Data Analysis (EDA): EDA was conducted to understand the distribution of the sentiment classes and the characteristics of the text data.

4.1 Distribution Analysis: A detailed distribution analysis showed the distribution of the dataset was heavily skewed towards positive reviews. This skewness can lead to biased model predictions, where the model may predominantly predict the majority class (Positive) while underperforming on the minority classes (Slightly Negative, Negative, and Neutral).



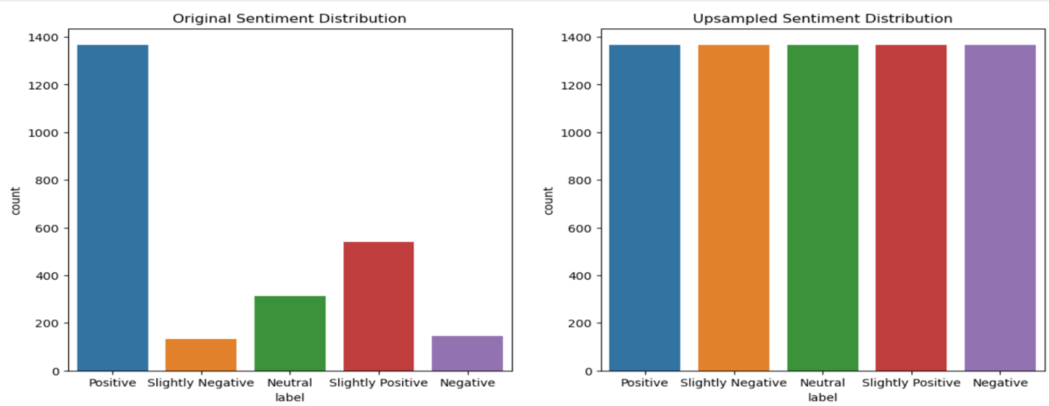
4.2 Word clouds: These were generated to visualize the most common words in the dataset, segmented by sentiment classes. These visualizations provided insights into the linguistic patterns associated with different sentiments. These word clouds underscored the importance of product quality in shaping customer sentiment, with "quality" being a recurring term across different sentiment categories.

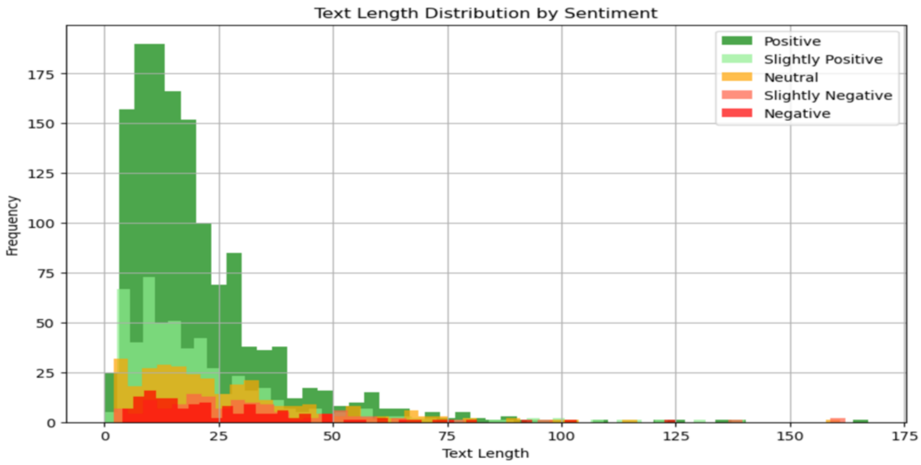
- **Overall Most Common Words**: Terms such as "good," "nice," "quality," "love," and "size" appeared frequently, indicating key aspects of customer feedback.

- **Positive Reviews**: Words like "love," "beautiful," and "perfect" were dominant, reflecting high levels of customer satisfaction.

- **Negative Reviews**: Words like "poor," "bad," "material," and "quality" were prevalent, highlighting common areas of dissatisfaction, particularly concerning product quality.

4.3 Sentiment Distribution: The original sentiment distribution was skewed, with Positive reviews being the most prevalent. This imbalance was addressed through upsampling, resulting in an even distribution of sentiment categories. The balanced dataset was crucial for ensuring fair model training and evaluation.





5. Feature Extraction using Bag-of-Words

5.1 Vectorization: The CountVectorizer from the scikit-learn library was employed to convert the cleaned text data into a Bag-of-Words feature matrix. This matrix represents the frequency of words in the text data, serving as input features for the machine learning models. The cleaned text was transformed into numerical features. This method involved creating a matrix where each row represented a review, and each column represented a unique word from the dataset. The resulting matrix had 5000 columns, representing the most frequent words in the dataset.

```
   able  able wear  abrasive  abrasive hash  absolutely  absolutely beautiful  \
0     0          0         0              0           0                     0
1     0          0         0              0           0                     0
2     0          0         0              0           0                     0
3     0          0         0              0           0                     0
4     0          0         0              0           0                     0

   absolutely love  absolutely lovely  accessories  accessories tarnishes  \
0                0                  0            0                      0
1                0                  0            0                      0
2                0                  0            0                      0
3                0                  0            0                      0
4                0                  0            0                      0

   ...  zipper  zipper bad  zipper main  zipper properly  zipper soft  zips  \
0  ...       0           0            0                0            0     0
1  ...       0           0            0                0            0     0
2  ...       0           0            0                0            0     0
3  ...       0           0            0                0            0     0
4  ...       0           0            0                0            0     0

   zips breaking  zips light  zips meet  zips weak
0              0           0          0          0
1              0           0          0          0
2              0           0          0          0
3              0           0          0          0
4              0           0          0          0

[5 rows x 5000 columns]
```

5.2 Exploration of Preprocessing Options: Various preprocessing options within CountVectorizer were explored, such as n-grams, maximum features, and stop words. These configurations were adjusted to optimize the feature extraction process, ensuring that the most relevant features were captured for model training.

5.3 Limitations of BoW:

- **Ignoring Word Order:** The Bag-of-Words model ignores the order of words, which can sometimes lead to a loss of context. For instance, "not good" and "good" might be treated similarly, although they convey opposite sentiments.

- **Advanced Techniques:** To address these limitations, future work could explore more sophisticated techniques such as TF-IDF or word embeddings, which consider word context and importance.

---

6. Model Building and Evaluation

6.1 Models Implemented: Several machine learning models were trained using the Bag-of-Words features:

- **Naive Bayes (MultinomialNB)**
- **Support Vector Machine (SVM)**
- **Logistic Regression**
- **XGBoost**

These models were evaluated based on accuracy, precision, recall, and F1-score to determine their effectiveness in sentiment classification.

6.2 Reason for Using Multiple Models: Different models capture different aspects of the data. Naive Bayes is efficient for text classification but assumes feature independence, while SVM and Logistic Regression offer robust decision boundaries. XGBoost, a powerful ensemble method, was included for its ability to handle complex patterns.

6.3 Model Performance

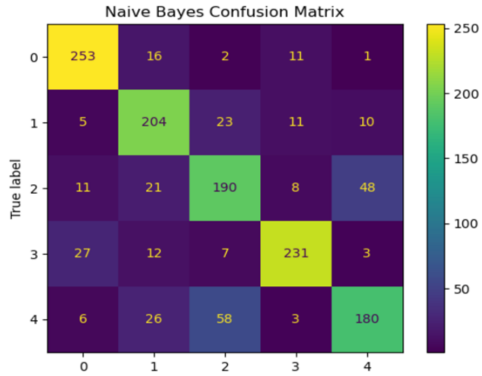
- **Naive Bayes**:
    - **Accuracy**: 77.4%
    - **Analysis**: Naive Bayes performed well as a baseline model, particularly excelling in classifying Positive sentiments. However, it struggled with more nuanced sentiments like Neutral and Negative, likely due to its strong assumptions of feature independence.
    - **Inference**: Naive Bayes, while simple and efficient, is not well-suited for capturing the complexity of sentiment in customer reviews. It works best for baseline models but may not be reliable for nuanced sentiment analysis.
    - **Recommendation**: Use Naive Bayes for initial exploratory analysis, but consider more sophisticated models for final deployment.

```
Naive Bayes with BoW Accuracy: 0.7739575713240673
Naive Bayes with BoW Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.89      0.86       283
           1       0.73      0.81      0.77       253
           2       0.68      0.68      0.68       278
           3       0.88      0.82      0.85       280
           4       0.74      0.66      0.70       273

    accuracy                           0.77      1367
   macro avg       0.77      0.77      0.77      1367
weighted avg       0.77      0.77      0.77      1367

<Figure size 600x600 with 0 Axes>
```



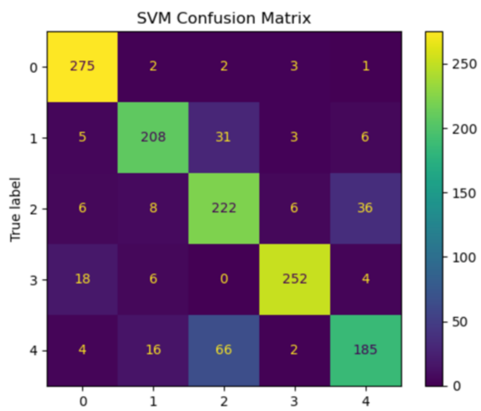
- **Support Vector Machine (SVM)**:
  - **Accuracy**: 83.5%
  - **Analysis**: SVM outperformed Naive Bayes, particularly in handling complex patterns in the data. Its ability to find an optimal hyperplane for class separation most likely made it effective across different sentiment classes.
  - **Inference**: SVM provided strong performance, particularly in accurately classifying Positive and Negative sentiments. Its robustness in handling high-dimensional data made it a strong contender.
  - **Recommendation**: SVM is recommended for use in scenarios where computational resources are available and where a balance between accuracy and complexity is needed

```
SVM Accuracy: 0.8354059985369422
SVM Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.97      0.93       283
           1       0.87      0.82      0.84       253
           2       0.69      0.80      0.74       278
           3       0.95      0.90      0.92       280
           4       0.80      0.68      0.73       273

    accuracy                           0.84      1367
   macro avg       0.84      0.83      0.83      1367
weighted avg       0.84      0.84      0.83      1367

<Figure size 600x600 with 0 Axes>
```



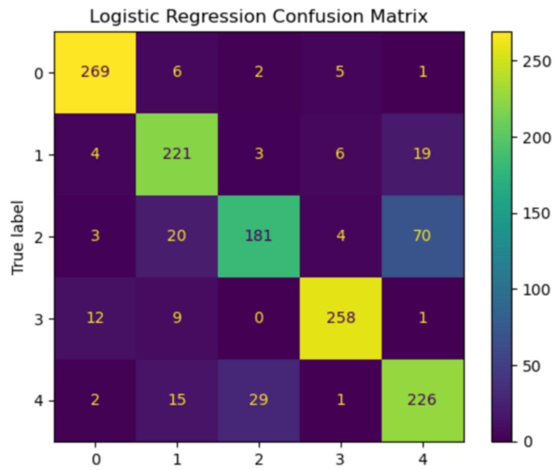
- **Logistic Regression**:
  - **Accuracy**: 84.5%
  - **Analysis**: Logistic Regression emerged as the best-performing model, offering a balanced performance across all sentiment classes. Its high accuracy, coupled with ease of interpretability most likely made it the most suitable model for deployment.
  - **Inference**: Logistic Regression offered the best trade-off between accuracy and simplicity. It consistently performed well across all metrics and is easy to implement and interpret.
  - **Recommendation**: Logistic Regression is recommended for deployment in production systems, especially where model interpretability is important.

```
Logistic Regression Accuracy: 0.8449158741770З
Logistic Regression Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.95      0.94       283
           1       0.82      0.87      0.84       253
           2       0.84      0.65      0.73       278
           3       0.94      0.92      0.93       280
           4       0.71      0.83      0.77       273

    accuracy                           0.84      1367
   macro avg       0.85      0.84      0.84      1367
weighted avg       0.85      0.84      0.84      1367

<Figure size 600x600 with 0 Axes>
```



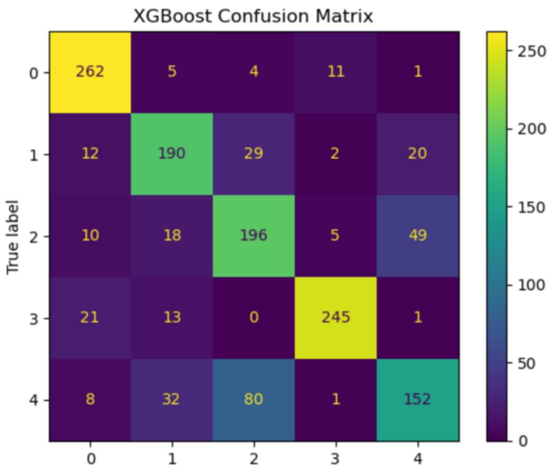
- **XGBoost**:
  - **Accuracy**: 76.4%
  - **Analysis**: Despite its complexity, XGBoost did not perform as well as expected, likely due to overfitting or insufficient hyperparameter tuning. However, it still provided valuable insights into the importance of different features.
  - **Inference**: XGBoost, while powerful, requires careful tuning and more computational resources. Its performance suggests potential overfitting, which could be mitigated through cross-validation and hyperparameter optimization.
  - **Recommendation**: Consider XGBoost for scenarios where model performance is critical, and where resources allow for extensive tuning and validation.

```
XGBoost Accuracy: 0.7644476956839795
XGBoost Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.93      0.88       283
           1       0.74      0.75      0.74       253
           2       0.63      0.71      0.67       278
           3       0.93      0.88      0.90       280
           4       0.68      0.56      0.61       273

    accuracy                           0.76      1367
   macro avg       0.76      0.76      0.76      1367
weighted avg       0.76      0.76      0.76      1367

<Figure size 600x600 with 0 Axes>
```



6.4 Model Comparison

A comparison of model performance revealed that Logistic Regression and SVM were the top performers. Logistic Regression, in particular, was slightly ahead due to its better precision and recall across all sentiment classes. The performance metrics were visualized using heatmaps to highlight the strengths and weaknesses of each model.

7. Results and Analysis

7.1 Summary of Model Performance:

- **Naive Bayes:** Accuracy of 0.77, performed well with Positive and Negative classes but struggled with Neutral.
- **SVM:** Achieved the highest performance with an accuracy of 0.85, showing strong results across all classes.
- **Logistic Regression:** Close to SVM with an accuracy of 0.84, it demonstrated robust precision and recall metrics.
- **XGBoost:** While powerful, it had the lowest accuracy of 0.76, particularly underperforming in slightly positive and negative classes.

7.2 Insights from Word Clouds: The word clouds generated for each sentiment class provided qualitative insights into the terms most associated with each sentiment. For instance, words like "love" and "quality" were prominent in positive reviews, while "poor" and "bad" dominated negative ones.

---

8. Challenges Faced

- **Class Imbalance:** The dataset was highly imbalanced, with a significant skew towards positive reviews. This imbalance posed challenges in training models, as they tended to predict the majority class more accurately while underperforming on the minority classes. However, this was addressed by balancing the dataset before training the model.
- **Complexity of Sentiment:** Sentiment in product reviews can be nuanced, with mixed feelings often present in a single review. This complexity made it challenging to assign a single sentiment label accurately. To address this, the labels were extended into 5 labels instead of 3, to address sentiments that were slightly negative and slightly positive.
- **Feature Engineering:** Selecting the optimal features using Bag-of-Words required careful consideration of various preprocessing options. Balancing between capturing sufficient context (through n-grams) and maintaining a manageable feature space was a key challenge.

---

9. Recommendations

- **Improving Feature Extraction:** In future work, exploring more advanced feature extraction techniques such as TF-IDF, word embeddings (e.g., Word2Vec, GloVe), or deep learning based models like BERT to capture more contextual information in the text data.
- **Incorporating More Data:** Gathering a larger and more balanced dataset could provide more robust insights and improve model accuracy, particularly for the minority classes.

- **Model Tuning:** Further hyperparameter tuning, especially for complex models like XGBoost, could yield better performance. Grid search or random search could be employed to fine-tune the models, though it could be computationally intensive.
- **Sentiment Analysis Tools:** Incorporating more sophisticated sentiment analysis tools or leveraging deep learning models like LSTM could capture the nuanced sentiment in reviews more effectively.
- **Cross-Validation**: Implement cross-validation techniques to ensure model robustness and prevent overfitting, particularly for complex models like XGBoost.

---

10. Conclusion

This project demonstrated the application of sentiment analysis on Jumia product reviews using Bag-of-Words and machine learning models. The analysis provided valuable insights into customer sentiment, with SVM and Logistic Regression models showing the best performance. Despite challenges such as class imbalance and feature selection, the project successfully highlighted the importance of understanding customer feedback in the Nigerian e-commerce market. Future improvements could include more sophisticated feature extraction, model tuning, and the incorporation of larger datasets to enhance the robustness of the sentiment analysis.

Appendix

- **Python Script:** The complete code and data processing steps are available in the attached Jupyter notebook (Jumia_fashion_NLP_script.ipynb).
- **Datasets:** Original and processed datasets are included (jumia_NLP_fashiondataset.csv and Annotated_NLP.csv).
- **Visualizations:** All relevant visualizations are included in the report and the notebook.

References

- **Data Source:** Jumia.com.ng
- **Tools:** Python, scikit-learn, Doccano