# 🧮 Sentiment Analysis on Jumia Smartphone Reviews Under 10K

This is an end to end project  combining  web scraping , SQL storage , NLP preprocessing, topic Modeling and Machine learning

## OBJECTIVE

To extract and analyze customer reviews from the Jumia e-commerce platform, focusing on smartphones priced under KES 10,000. The goal was to uncover customer sentiment, identify product strengths and weaknesses, and evaluate both retailer and product performance.

🧰 Tools & Technologies Used

| CATEGORY | TOOLS/ Technologies |
|---|---|
| Web Scraping | Scrapy, Selenium |
| Data storage | SQLite |
| Data preprocessing | Pandas, Sqlite3, re |
| NLP – Sentiment analysis | NLTK, Textblob, Varder, Roberta |
| Topic Modeling | LDA (Latent Dirichlet Allocation) |
| Feature Engineering | TF-IDF (Unigrams & Bigrams) |
| Machine Learning | Scikit-learn (Naive Bayes, Logistic Regression, SVM) |
| Visualization | Seaborn ,Matplotlib |
| Environment | VSCode, Jupyter Notebook ,Python |

PROJECT WORKFLOW

**1. Data Extraction**

- Scraped reviews from multiple Jumia product pages.
- Used **Scrapy** to collect phone URLs, and **Selenium** to navigate dynamic review pages.

- Extracted: product metadata, overall rating, individual reviews, star ratings, and dates.

**2. Data Storage**

- Stored raw data in **SQLite**, using two linked tables:
    - **Product Table**: phone ID, name, price ,discount, brand, overall reviews .
    - **Review Table**: phone ID, review text, date, star rating.

**3. Data Preprocessing**

- Cleaned text: lowercased, removed special characters and stopwords.
- Applied **tokenization and lemmatization** for consistency.
- Even short and empty reviews were included to evaluate their impact on ML models.

💬 **Sentiment Labeling**

- Applied **TextBlob**, **VADER**, and **RoBERTa** to generate sentiment labels.
- Compared results with star ratings (5–4 = Positive, 3 = Neutral, 2–1 = Negative).
- Found **12.5% mismatch** between sentiment scores and star ratings, often due to sarcasm, misused star ratings, or ambiguous language.

🧠 **Topic Modeling**

- Used **LDA** to uncover top 3 themes in reviews:
    1. **Jumia Services**
    2. **Customer Satisfaction**
    3. **Phone Features**

⚙️ **Feature Engineering**

- Vectorized text using **TF-IDF (unigrams + bigrams)**.
- Included features: sentiment scores, topic tags and star rating.

Modeling

Split the dataset into training and testing sets (80/20)

Trained and compared the following classifiers:

Naïve Bayes
balanced classification SVM

SMOTE (Synthetic Minority Oversampling Technique)  SVM

| MODEL | Accuracy | Precision | Recall | F1- Score |
|---|---|---|---|---|
| Logistic regression | 0.80 | | | |
| Naive Bayes | 0.82 | 0.75 | 0.82 | 0.78 |
| Balanced SVM | 0.68 | 0.78 | 0.68 | 0.71 |
| SMOTE- biased SVM | 0.74 | 0.74 | 0.75 | 0.74 |

🎯 **Goal Focus**: Achieving high **recall for negative reviews**. Although Naive Bayes performed well overall, Balanced SVM was better at catching negatives.

# 📈 Key Insights

- **70% of reviews were Positive, 22% Negative, 8% Neutral.**
- **Battery, charging, camera, and price were the most mentioned features.**
- **RoBERTa showed the highest agreement with human star ratings among sentiment models (65.6% accuracy), followed by VADER.**
- **Nokia had the most polarized reviews (strongly positive and negative).**
- **Itel had the highest number of low-budget phones (<10K).**
- **Customer Satisfaction was the most frequent topic, with over 82% positive sentiment.**
- **Phone Features drew the highest negative feedback.**
- **Jumia Services received the least attention (18% of total reviews).**

⚠️ **Challenges**

- Class imbalance: Most reviews were positive, making it hard to train models on negative/neutral classes.
- Customers misused star ratings (e.g., 5-star reviews with complaints).

- Sarcasm and humor were common and difficult for sentiment tools to interpret.
- SMOTE overcompensated, making the model biased toward negative predictions.

🚀 **Next Steps & Improvements**

- Expand scraping to **other platforms** (e.g., Kilimall) for comparative insights.
- Add more categories (e.g., tablets, accessories) to widen the scope.
- Collect a **larger dataset** for better training and balanced sentiment distribution.
- Explore **transformer-based models** (like fine-tuned RoBERTa or BERT) for more context-aware sentiment classification.

## PROJECT REPOSITORY

GitHub: https://github.com/Adieltheanalyst/jumia_sentiment_analyser

💡 **What I Learned**

- Building robust web scraping pipelines with Scrapy + Selenium.
- Using SQL to structure and store scraped data.
- Applying and evaluating multiple sentiment analysis techniques.
- Using **LDA** for unsupervised topic discovery.
- Training and tuning ML models with an **imbalanced dataset**.
- Communicating technical insights in a clear and structured format.

By Adiel Maina