FLIPKART SENTIMENTAL ANALYSIS

SUBMITED BY
AUSTIN ROMEO A

Table of contents

Project Overview:

The goal of this project to scrape product data and reviews from Flipkart, clean and process the data, perform sentiment analysis, and recommend products using LangChain based on customer reviews. The project is deployed on AWS to ensure scalability and accessibility.

1. Web Scraping Mobile Phone Data

- **Objective:** Scrape the product name, price, and product link for five mobile phones across multiple pages on Flipkart.
- **Tools:** Python libraries (BeautifulSoup, requests, selenium).
- **Steps:**
  - i. Select five mobile phone product categories on Flipkart.
  - ii. Write a script to scrape product names, prices, and links.
  - iii. Handle pagination to scrape data from multiple pages.
  - iv. Save the extracted data into a CSV file with columns: Product Name, Price, and Link.

2. Data Cleaning and Filtering by Price

- **Objective:** Clean the scraped data and filter the price column to retain only phones priced between ₹20,000 and ₹40,000.
- **Tools:** Pandas
- **Steps:**
  - i. Convert the price column to numerical format (remove currency symbols, commas).
  - ii. Filter the data to include only phones with prices between ₹20,000 and ₹40,000.
  - iii. Save the cleaned data into a new CSV file.

3. Scraping Reviews and Star Ratings

- **Objective:** Using the product links, scrape the reviews and star ratings for each phone.
- **Tools:** BeautifulSoup, requests, selenium

- **Steps:**
    1. For each product link, navigate to the reviews section.
    2. Scrape the user reviews and star ratings.
    3. Handle pagination to collect reviews from multiple pages.
    4. Store the scraped data (Product Name, Reviews, Star Ratings) in a CSV file.

4. Data Cleaning for Reviews

- **Objective:** Clean the review text to remove unwanted characters and format the data for analysis.
- **Tools:** Pandas, re (regular expressions)
- **Steps:**
    1. Remove symbols, commas, and apostrophes from the reviews.
    2. Convert all review text to lowercase.
    3. Remove emojis from the reviews.
    4. Implement an auto-correction method to fix spelling errors.
    5. Save the cleaned data into a CSV file.

5. Combining CSV Files

- **Objective:** Merge all the scraped and cleaned data into a single CSV file, excluding certain columns.
- **Tools:** Pandas
- **Steps:**
    1. Merge the product and review data.
    2. Drop the product link and price columns from the combined dataset.
    3. Save the final combined dataset

6. Converting Product Name to Product ID

- **Objective**: Convert product names into unique product IDs for easier analysis.
- **Tools:** Pandas
- **Steps:**
    i. Use pd.factorize() to assign a unique numerical ID to each product name.
    ii. Replace the product name with the corresponding product ID in the final dataset.
    iii. Save the updated dataset.

7. Sentiment Analysis for Product Recommendation

- **Objective:** Perform sentiment analysis on the product reviews and assign sentiment scores.
- **Tools:** VADER, TextBlob, or NLTK
- **Steps:**
    i. Use a sentiment analysis library to evaluate the polarity of each review (positive, negative, neutral).
    ii. Assign sentiment scores based on the review text.
    iii. Aggregate the scores to provide an overall sentiment score for each product.

8. Exploratory Data Analysis (EDA)

- **Objective**: Conduct EDA on the cleaned data and sentiment analysis results to gain insights into product performance, customer sentiment, and patterns in the data.
- **Tools:** Pandas, Matplotlib, Seaborn.
- **Steps:**
    a. Distribution Plots:
        i. Plot the distribution of star ratings and sentiment scores to see overall customer satisfaction.
    b. Bar Charts:
        i. Create bar charts to visualize the top 10 highest-rated products based on star ratings.
        ii. Visualize the products with the highest and lowest sentiment scores.

c. **Correlation Analysis:**

   i. Use correlation heatmaps to analyze the relationship between price, sentiment score, and star ratings.

d. **Sentiment Trend:**

   i. Plot a trend of the sentiment scores for different products to identify which products consistently receive positive or negative feedback.

e. **Word Cloud:**

   i. Generate a word cloud to visualize the most frequently used words in the reviews (after cleaning).

9. Product Recommendation System using Langchain

- **Objective:** Develop a product recommendation system based on the sentiment analysis and user queries using Langchain.
- **Tools:** Langchain
- **Steps:**

  i. Set up Langchain to handle user queries.

  ii. Integrate the sentiment scores and product data to generate personalized recommendations.

  iii. Ensure the system recommends products within the specified price range and sentiment score.

10. Deployment on AWS

- **Objective:** Deploy the final product recommendation system on AWS for scalability and accessibility.
- **Tools:** AWS (EC2, Lambda, S3, RDS)
- **Steps:**

  i. Set up an AWS EC2 instance or AWS Lambda function to host the application.

  ii. Use S3 to store the scraped and cleaned data.

  iii. Set up RDS (Relational Database Service) if required for persistent storage.

  iv. Deploy the application and expose it via a REST API or web interface for users.

Conclusion

By following these steps, the project will efficiently scrape product data, perform sentiment analysis, and recommend products, ultimately deploying the solution on AWS. Each step ensures that the data is handled and cleaned properly for accurate recommendations.

Final Deliverables:

1. **Scraped Data**: Product names, prices, reviews, and ratings from Flipkart.
2. **Cleaned Data**: Filtered reviews and structured data ready for analysis.
3. **Sentiment Analysis Results**: Sentiment classification of customer reviews.
4. **Product Recommendations**: Top products recommended based on sentiment analysis using LangChain.
5. **Visualizations**: EDA graphs showing sentiment and rating trends.
6. **AWS Deployment**: A live system deployed on AWS for real-time product recommendation

References :

1.Pandas Documentation

2. Selenium Documentation

Data visualization :

- Seaborn Documentation
- Matplotlib Documentation
- Plotly Documentation
- Sentimental Analysis Documentation
- Langchain Documentation
- AWS Documentation