

Sentiment Analysis API Documentation

Submitted by

SYED HARIS ALI

Table of Contents

1. Introduction
2. Requirements
3. Architecture
4. Setup and Installation
5. Code Explanation
 - 1. Importing Libraries
 - 2. Setting Up Azure Credentials
 - 3. Creating the Flask Application
 - 4. Defining the Blob Download Function
 - 5. Defining the Sentiment Analysis Function
 - 6. Performing Sentiment Analysis on Reviews
 - 7. Defining the Index Route
 - 8. Running the Flask Application
6. API Endpoints
7. Testing the API
8. Results
9. Conclusion
10. References

1. Introduction

This document outlines the development of a Sentiment Analysis API built with Flask and Azure Cognitive Services. The API retrieves customer reviews stored in Azure Blob Storage, performs sentiment analysis on them, and provides a web interface to display the results.

2. Requirements

Software Requirements:

- Python 3.x
- Flask
- Pandas
- Azure Storage Blob SDK
- Requests library

Azure Requirements:

- Azure Blob Storage Account
- Azure Cognitive Services Text Analytics Account

Installation of Required Libraries:

OS.

Pandas

Flask

Azure

Requests

3. Architecture

The architecture consists of:

- Client: Sends requests to the Flask API.
- Flask API: Handles requests, processes data, and returns responses.
- Azure Blob Storage: Stores customer review files (CSV and Excel).
- Azure Cognitive Services: Provides sentiment analysis functionality.

4. Setup and Installation

Azure Setup:

- Create an Azure account.
- Set up Blob Storage and upload the review files (CSV/XLSX).
- Create a Text Analytics resource and obtain the endpoint and API key.

Code explanation

1. Import necessary libraries for file handling, web application, data manipulation, Azure integration, and HTTP requests.

2. Setting Up Azure Credentials

Configure Azure storage and Cognitive Services credentials.

3. Creating the Flask Application

Initialize the Flask application.

4. Defining the Sentiment Analysis Function

Calls Azure's Text Analytics API to analyze sentiment based on the provided review text.

5. Performing Sentiment Analysis on Reviews

Downloads review files and performs sentiment analysis on the review column, returning a DataFrame with results.

6. Defining the Index Route

Defines the home route, retrieves sentiment analysis results, and renders them in an HTML template.

7. Running the Flask Application

Runs the application in debug mode for development.

Refer the app.py file

6. API Endpoints

Root Endpoint

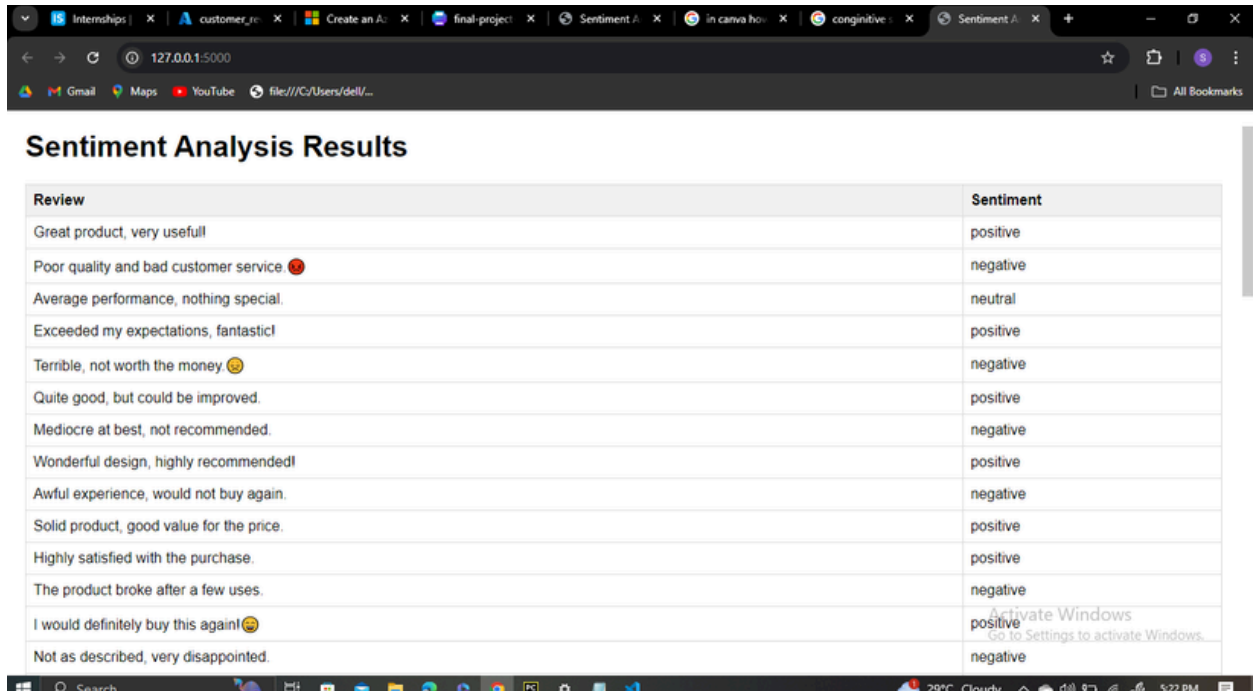
- URL: /
- Method: GET
- Description: Renders the index page with the sentiment analysis results.
- Response: Returns an HTML page displaying reviews and their corresponding sentiments.

7. Testing the API

Access the Application: Open a web browser and navigate to <http://127.0.0.1:5000/>.

Check Results: Ensure customer reviews are displayed along with their sentiment analysis.

8. Results



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000'. The page title is 'Sentiment Analysis Results'. Below the title is a table with two columns: 'Review' and 'Sentiment'. The table contains 15 rows of data. The reviews are: 'Great product, very usefull', 'Poor quality and bad customer service', 'Average performance, nothing special.', 'Exceeded my expectations, fantastic!', 'Terrible, not worth the money', 'Quite good, but could be improved.', 'Mediocre at best, not recommended.', 'Wonderful design, highly recommended!', 'Awful experience, would not buy again.', 'Solid product, good value for the price.', 'Highly satisfied with the purchase.', 'The product broke after a few uses.', 'I would definitely buy this again!', and 'Not as described, very disappointed.' The sentiment classifications are: positive, negative, neutral, positive, negative, positive, negative, positive, negative, positive, positive, negative, positive, and negative. There is a watermark 'Activate Windows' on the right side of the table.

Review	Sentiment
Great product, very usefull	positive
Poor quality and bad customer service	negative
Average performance, nothing special.	neutral
Exceeded my expectations, fantastic!	positive
Terrible, not worth the money	negative
Quite good, but could be improved.	positive
Mediocre at best, not recommended.	negative
Wonderful design, highly recommended!	positive
Awful experience, would not buy again.	negative
Solid product, good value for the price.	positive
Highly satisfied with the purchase.	positive
The product broke after a few uses.	negative
I would definitely buy this again!	positive
Not as described, very disappointed.	negative



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000'. The page title is 'Sentiment Analysis Results'. Below the title is a table with two columns: 'Review' and 'Sentiment'. The table contains 15 rows of data. The reviews are: 'Not as described, very disappointed.', 'Customer service was really helpful.', 'It works perfectly as expected.', 'I had a terrible experience with this.', 'Well worth the money.', 'Waste of time and money.', 'Absolutely love this product!', 'It didn't work as advertised.', 'Very reliable and easy to use.', 'I had higher expectations, sadly disappointed.', 'Delivered quickly, works as intended.', 'Horrible quality, won't buy again.', 'This is the best purchase I've made.', 'It stopped working after a month.', 'I'm quite pleased with this product.', 'Very cheap material, not durable.', 'Fantastic performance for the price.', and 'Extremely frustrating experience.' The sentiment classifications are: negative, positive, positive, negative, positive, negative, positive, negative, positive, negative, positive, negative, positive, negative, positive, negative, positive, and negative. There is a watermark 'Activate Windows' on the right side of the table.

Not as described, very disappointed.	negative
Customer service was really helpful.	positive
It works perfectly as expected.	positive
I had a terrible experience with this.	negative
Well worth the money.	positive
Waste of time and money.	negative
Absolutely love this product!	positive
It didn't work as advertised.	negative
Very reliable and easy to use.	positive
I had higher expectations, sadly disappointed.	negative
Delivered quickly, works as intended.	positive
Horrible quality, won't buy again.	negative
This is the best purchase I've made.	positive
It stopped working after a month.	negative
I'm quite pleased with this product.	positive
Very cheap material, not durable.	negative
Fantastic performance for the price.	positive
Extremely frustrating experience.	negative

Conclusion

This documentation has outlined the development of a sentiment analysis API using Flask and Azure services. The API successfully retrieves reviews from Azure Blob Storage, processes them using Azure's Text Analytics, and presents the results through a web interface. Future enhancements may include adding authentication, more detailed analysis, and handling various languages.