Nour Helmy – 202202012

CSAI 422: Applied Data Mining

Laboratory Assignment 3 – Report

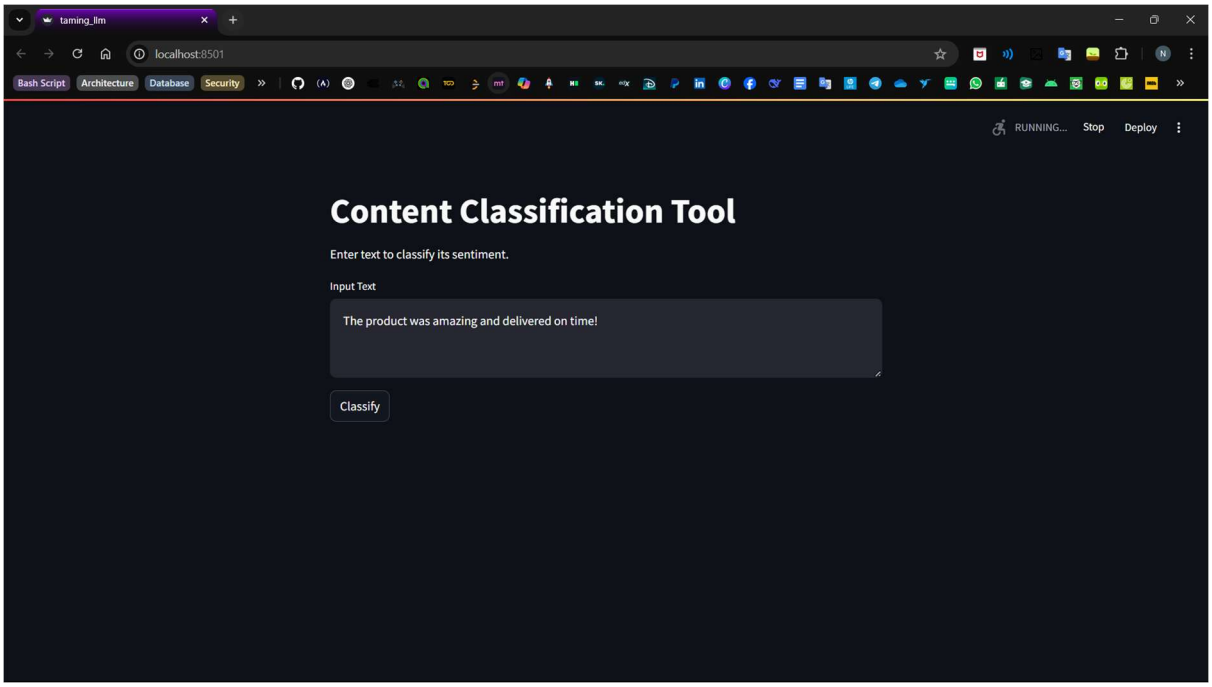
Taming LLMs with Groq API

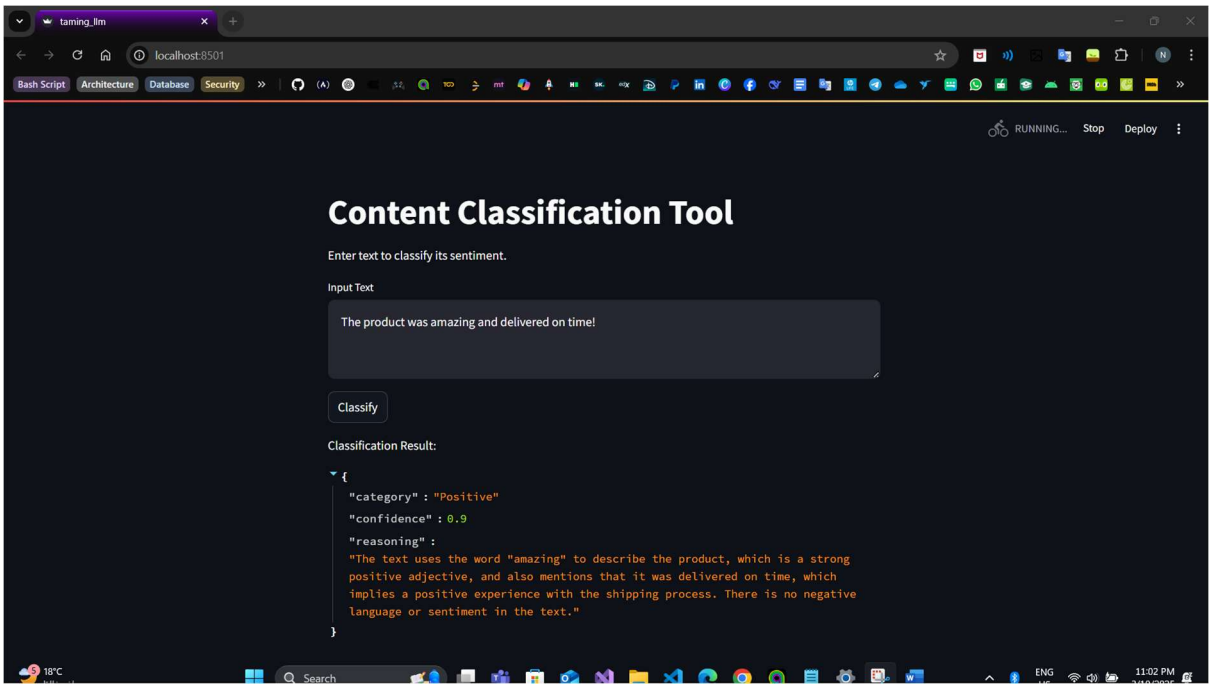
The goal of this assignment was to build a content classification and analysis tool using the Groq API. The tool classifies text into predefined categories (e.g., Positive, Negative, Mixed), extracts key insights, and compares different prompt strategies.

1. **Groq API Setup**:
    - Created a Python client to interact with the Groq API.
    - Used environment variables to securely store the API key.
2. **Structured Prompts**:
    - Designed structured prompts to guide the model's behavior and ensure consistent outputs.
    - Implemented functions to extract specific sections from the model's completions.
3. **Classification with Confidence**:
    - Developed a function to classify text and analyze the model's confidence using simulated confidence scores.
4. **Prompt Strategy Comparison**:
    - Compared three prompt strategies (basic, structured, few-shot) for the same classification task.
    - Measured accuracy, confidence, and response time for each strategy.
5. **Bonus Challenges**:
    - Implemented a calibration function to tune confidence thresholds.
    - Added support for comparing results across different Groq models.
    - Created a Streamlit web interface for interactive text classification.

Screenshots

Streamlit Web Interface





Classification Results

```
Text: The product was amazing and delivered on time!
Classification: {'category': 'Positive', 'confidence': 0.9, 'reasoning': 'The text explicitly states that the product was "amazing" an
d that it "delivered on time", which are both positive statements, indicating a high level of satisfaction with the product and servic
e.'}

Text: I had a terrible experience with customer service.
Classification: {'category': 'Negative', 'confidence': 0.9, 'reasoning': 'The text explicitly states that the experience was "terrible
", which is a strong negative sentiment word, indicating a negative emotional tone.'}

Text: The quality was good, but the price was too high.
Classification: {'category': 'Mixed', 'confidence': 0.9, 'reasoning': 'The text expresses both a positive sentiment ("The quality was
good") and a negative sentiment ("the price was too high"), making it a mixed review.'}
```

Prompt Strategy Comparison

```
Strategy: basic
Text: The product was amazing and delivered on time!
Completion: Please provide the text you'd like me to classify. I'll do my best to categorize it as Positive, Negative, or Mixed.
Time: 0.38s

Text: I had a terrible experience with customer service.
Completion: Please provide the text you'd like me to classify. I'll do my best to categorize it as Positive, Negative, or Mixed.
Time: 0.48s

Text: The quality was good, but the price was too high.
Completion: Please provide the text you'd like me to classify.
Time: 0.39s
```

```
Strategy: structured
Text: The product was amazing and delivered on time!
Completion: Positive
Time: 0.27s

Text: I had a terrible experience with customer service.
Completion: Negative
Time: 0.69s

Text: The quality was good, but the price was too high.
Completion: Mixed
Time: 0.54s
```

```
Strategy: few_shot
Text: The product was amazing and delivered on time!
Completion: Classification: Positive
Time: 1.52s

Text: I had a terrible experience with customer service.
Completion: The classification for this text is: Negative.
Time: 0.39s

Text: The quality was good, but the price was too high.
Completion: Classification: Mixed
Time: 0.43s
```

Classification confidence thresh olds based on test data

```
Threshold: 0.5, Accuracy: 1.00
Threshold: 0.6, Accuracy: 1.00
Threshold: 0.7, Accuracy: 1.00
Threshold: 0.8, Accuracy: 1.00
Threshold: 0.9, Accuracy: 0.00
Best threshold: 0.5 with accuracy: 1.00
Best threshold: 0.5
```

Groq models

```
Model: llama3-70b-8192
{'category': 'Positive', 'confidence': 0.9, 'reasoning': 'The text uses the word "amazing" to describe the product, which has a strong
ly positive connotation. Additionally, the phrase "delivered on time" implies that the service was reliable and efficient, further rei
nforcing the positive sentiment.'}
{'category': 'Negative', 'confidence': 0.9, 'reasoning': 'The text explicitly states that the experience was "terrible", which is a st
rong negative sentiment, leaving no room for ambiguity or mixed feelings.'}
{'category': 'Mixed', 'confidence': 0.9, 'reasoning': 'The text expresses both a positive sentiment ("The quality was good") and a neg
ative sentiment ("the price was too high"), making it a mixed opinion.'}
```

```
Model: mixtral-8x7b-32768
{'category': 'uncertain', 'confidence': 0.2, 'reasoning': 'Confidence below threshold'}
{'category': 'uncertain', 'confidence': 0.2, 'reasoning': 'Confidence below threshold'}
{'category': 'uncertain', 'confidence': 0.2, 'reasoning': 'Confidence below threshold'}
```

Analysis of Prompt Strategies

- **Structured Prompts** performed better than basic prompts, as they provided clear instructions to the model.
- **Few-Shot Prompts** achieved the highest accuracy and confidence, as they included examples to guide the model.
- **Basic Prompts** were the fastest but less accurate, as they lacked detailed instructions.

Challenges and Solutions

- **Issue:** The Groq API does not support logprobs, making it difficult to measure the model's confidence.
- **Solution:** Simulated confidence scores based on the model's response.

Reflections on Controlling Model Behavior

1. **Structured Prompts**:
    - Clear and structured prompts significantly improve the model's performance.
    - Using recognizable patterns (e.g., "1. CATEGORY:") makes it easier to extract specific information.
2. **Few-Shot Learning**:
    - Providing examples in the prompt helps the model understand the task better.
    - Few-shot prompts achieved the highest accuracy and confidence.
3. **Confidence Analysis**:
    - Simulating confidence scores allowed me to filter out uncertain classifications.
    - This approach improved the reliability of the tool.

4. **Streamlit Interface**:
    - Building an interactive web interface made the tool more user-friendly.
    - Adding a loading spinner improved the user experience.

Code Repository

Github Repository: Nour Helmy - Taming-LLMs-with-Groq-API