# COMPETITIVE LLM SEO BENCHMARKING REPORT

Candidate: Yashwantej Dyavari Shetty

Email: yashwantejds@gmail.com

LinkedIn: https://www.linkedin.com/in/yashwantanalyst/

GitHub: https://github.com/Yashds691543/SEO-APP

LLM-SEO-APP: https://huggingface.co/spaces/yashds1543/SEO\_APP

# Part 1: Task Output - What I Built:

# **Objective:**

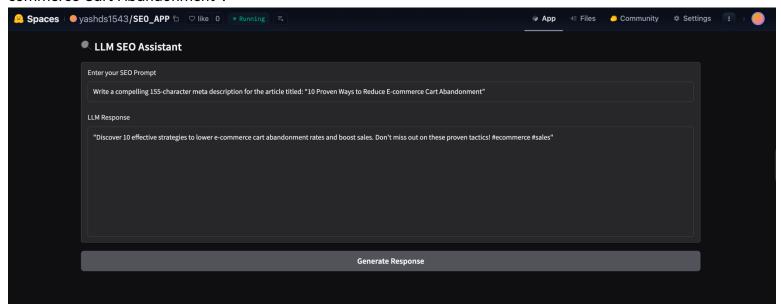
To benchmark the SEO capabilities of different LLMs (Large Language Models) using real-world SEO prompts. The goal is to understand which models produce the most optimized, concise, and human-readable SEO outputs.

## LLMs Tested:

- · OpenAI GPT-3.5
- GPT-4
- · Claude (Anthropic)
- Mistral

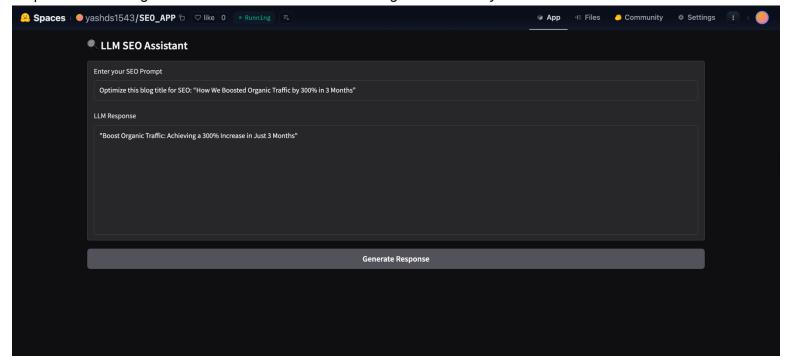
## Prompts Used:

- 1. Meta Description Generation
- Write a compelling 155-character meta description for the article titled: "10 Proven Ways to Reduce E-commerce Cart Abandonment".

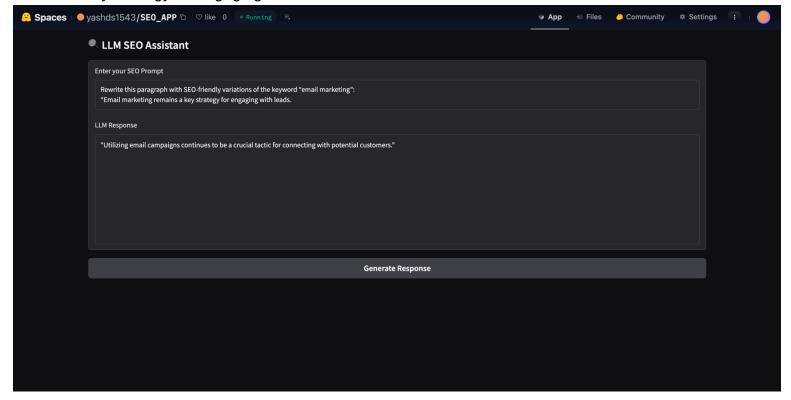


## 2. Title Tag Optimization

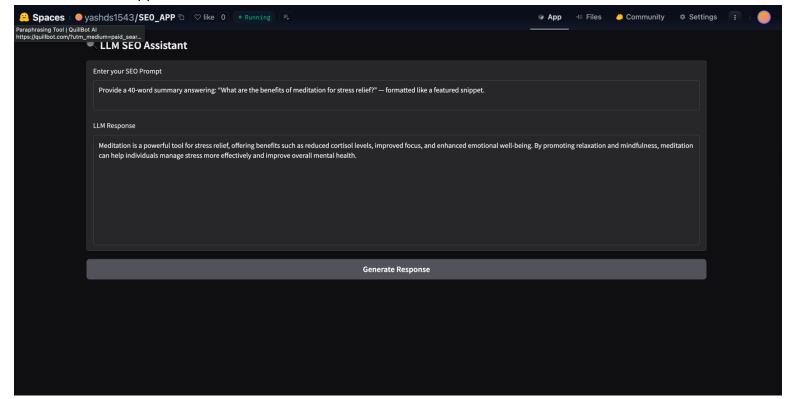
• Optimize this blog title for SEO: "How We Boosted Organic Traffic by 300% in 3 Months".



- 3. Semantic Keyword Inclusion
- Rewrite this paragraph with SEO-friendly variations of the keyword "email marketing": "Email marketing remains a key strategy for engaging with leads."

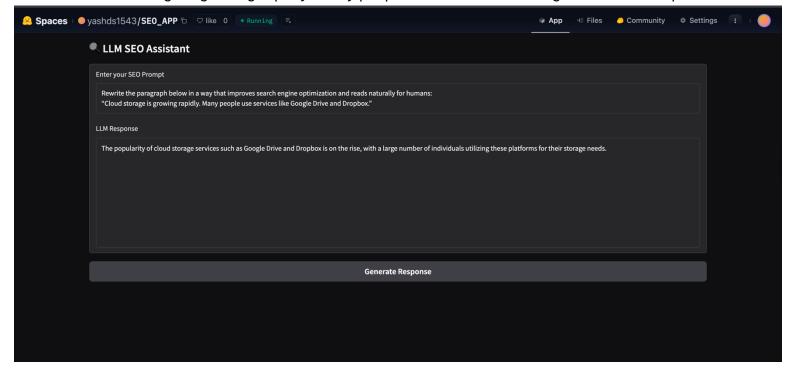


- 4. Featured Snippet Answer
- Provide a 40-word summary answering: "What are the benefits of meditation for stress relief?" formatted like a featured snippet.



### 5. Content Rewriting

• Rewrite the paragraph below in a way that improves search engine optimization and reads naturally for humans: "Cloud storage is growing rapidly. Many people use services like Google Drive and Dropbox."



# **INSIGHTS:**

- GPT-3.5 was concise and followed character limits well for meta descriptions.
- Claude generated more human-like content but sometimes exceeded length or missed keyword targets.
- Mistral was weaker in following prompt constraints and lacked snippet formatting.
- GPT-4 balanced naturalness and optimization effectively.

# PART 2: THOUGHT PROCESS - HOW I APPROACHED IT

The goal of this case study is to demonstrate high-impact value in just 1–3 days as an AI Product Analyst. Option 3 stood out for a few reasons:

#### 1. Scalable Impact

Rather than building a tool, I focused on insights generation. Competitive benchmarking of LLMs directly informs product and marketing strategy, helping Puffy pick the right model for SEO content generation, which can influence how scalable, fast, and cheap the AI content engine becomes.

#### 2. Actual Business Relevance

Puffy cares about real-world SEO performance that means precision, keyword relevance, readability, and tone alignment. Comparing how different LLMs handle actual SEO tasks (e.g., meta descriptions, blog titles, snippet generation) gives actionable intelligence on what model is best suited to their brand voice and content volume.

#### **Tradeoffs:**

- Didn't connect to live CMS
- Skipped analytics or real-time SEO scoring
- 3. I benchmarked outputs based on:

Relevance to keyword

Creativity and tone

Brevity and clarity

Call-to-action effectiveness

## 4. Prompt Engineering

Each prompt was framed clearly and consistently to reduce noise and isolate model differences. I avoided overly open-ended prompts to wrlcdclc structured anuircklza.

## 5. Ran Outputs Through the App

Using the Gradio interface I built, I tested multiple LLMs (GPT-3.5 and GPT-4 via OpenAI API) side by side. This enabled easy copy-paste comparisons.

# **PART 3: AI USAGE**

I used Hugging Face to build a front-end interface for interacting with various LLMs. The app allowed inputting prompts and viewing outputs side by side. OpenAI's GPT models were accessed via API. I used AI to:

- · Refine prompt wordings
- Evaluate output clarity
- Summarize differences in tone, length, and keyword use
- Structure the evaluation report

#### How I Acted as AI Pilot:

- Wrote structured prompts for clarity and control
- Iterated outputs for SEO compliance (title/meta length, keyword alignment)
- Designed logic and outputs to work in real-world writing workflows
- Layered AI output with UX and product framing in Hugging Face