Project Name

Sentiment Analysis API for Product and Service Reviews

Team with Contact information

Jackson Aden, 435-512-8748, jacksonaden@gmail.com,

Project Purpose

This is designed to be used for Reputation Management purposes for different businesses. They'll be able to call my API, send a batch with all of their reviews, and they'll receive a batch of sentiment scores for each review. They will have an account (backend setup), and get charged per API usage.

Background/Prior Knowledge

I worked for a company that did something similar, but using a generative AI model and with direct integration into their product. I did lots of research in rebuilding the product using a smaller, but just as effective model. I am a novice, having done lots of research hours, and making different AI prototypes of a similar nature. I still feel like there could be more that can be done, and I might even consider refining, or tuning, a small scale model to specific reviews. It would be harder, but I feel like that's where I sit.

There aren't a lot of very popular versions of this product, but there are other things that are similar. I'll be building this from scratch, but I'll definitely use some references from open source code and resources to make sure I'm optimizing the product.

I'm refining a similar project that I have made before, but with more freedom to use other tools and processes, without company oversight, and with more general usage. Instead of making just a prototype, I want to be making something that's production ready. I'll specifically be focusing more on building an API, a key generator, and a tracker for that key.

I really enjoy using generative AI tools, but I want them to do everything super well. I want AI to read my textbook, and then tell me what I'm looking for, without key word search. I want it to be able to read a book and tell me whether or not it's appropriate. These outcomes are so unmeasurable with generative text models, which are designed to give a different outcome with every input. There are open source ML models that I've messed with that I've really enjoyed, and sentiment analysis is a very unique and not often used one.

Description

From a user perspective, they will create an account, which will generate an API key. This API key will be part of the API call they make to my sentiment analysis. Depending on how many reviews they send in, then a score will be added to that specific key, which will later be used to charge a card for usage. The actual api itself, will send a review, or statement, about a product or service, returning a simple integer describing whether the review was positive, negative, or neutral. This is using a modified hugging face ML model (specifically _____) that vectorizes and analyzes the data, then shooting out the integer. This can be used to graph or show the overall sentiment based purely on reviews, or can be used to graph sentiment depending on time period, etc...

The intent is for the product to be used in a reputation management context. Maybe, in the future, I'll make a website for small businesses to do basic reputation management data analysis, graphing changes in sentiment over time etc... This project will make that way easier in the future, or for other companies to be able to do something similar. The intended audience could be anyone wanting to develop sentiment solutions, or small businesses wanting to see their true sentiment. I'll see if auto scaling is a part of the hosting solutions I use, but it's not a requirement, so it

This is specifically for companies that already use AI for customer reputation management. There are a few that could benefit from an API like this.

The project will be done when I can use an API key that I've generated, to call a Rest API, and receive the sentiment analysis. Then, I'll be able to go behind the scenes, and see that a certain API was used that many times. I'll also be able to send a batch, and that same behind the scenes tactic will show every time the software was used by that key.

Significance

This will give the opportunity to show that a tool like this is simple to use, simple to monetize, and simple to upkeep. This project will also show how inexpensive and accurate using other ML tools over generative AI can be.

This is also absolutely a project I can throw on my resume. It would not only show that I am willing to use newer technologies like AI, but will also show that I know how to find the right tools for the right job. Being able to explain to a hiring manager how I looked for a cheap alternative to using Chat GPT with the same performance, if not better, ears will perk up.

New Computer Science Concepts

I will need to learn how to create an API key, how to track the usage, and how to host my program. Learning to host a program on something like AWS or GCP is

a DevOps skill that can be super impressive, and could help me pivot into that as a career. I want to be a software architect, so getting familiar with these tools is important. It will be the majority of the time I spend on the project. Api's are constantly used in software too, so that if one program goes down, we can trace it to the piece that's broken, and the whole program won't go down because of it.

Interestingness

This project is something I feel like I can really have nailed down, and feel good about. I'll mostly be motivated by the accomplishment of finishing this project. I also really enjoy AI as a concept. I want to build a personal virtual assistant one day, and something like this would begin that process.

Milestones, Tasks and Schedule

Base Milestones with Estimated Time

- Requirements Elicitation (3-4 hours)
- Choose ML model (1-2 hours)
- Create API (3-4 hours)
- Make a user account creator (3-4 hours)
- Make an API key Generator (1-2 hours)
- Have the api call send some code to something else to keep track (DB?) (4-5 hours)
- Hook API to ML model (2-3 hours)
- Format Json input/output (1-2 hours)
- Create Tests for inputting and outputting (1-2 hours)
- Finish Prototype (5-6 hours)
- Host working code onto Backend as a Service (4-5 hours)
- Make API forward facing (1-2 hours)

Stretch Goals with Estimated Time

- Website for Sign up (10-12 hours)
- Connection to charging accounts (10-12 hours)
- Google Maps API compatibility (10-12 hours)
- .exl and .csv compatibility (5-6 hours)
- Train ML models on review data (9-10 hours)

Week by week plan

Week 3 - Finish Proposal and setup development

Week 4 - Research and setup Machine Learning model locally

Week 5 - Api Research and Development

Week 6 - Finish and Optimize API

Week 7 - Create the AI tool

Week 8 - Create user account creator, and research api key tracking

Week 9 - Create DB, Hook it up to the AI tool

Week 10 - Finalize full integration, api, tool, all locally (finish prototype)

Week 11- Fully implement testing files

Week 12 - Format Json, research and decide on Hosting Tools

Week 13 - Host AI, Tool, and API in production

Week 14 - Make the API able to be called by anyone, and do User Acceptance testing

Week 15 - Begin Website Development, (work with graphic design friend to make it look good)

Week 16 - Finish Website (get it up and User Acceptance Tested)

Week 17 - Create account creating software

Week 18 - Setup API card charges

Week 19 - Research and design .csv and .exl feature

Week 20 - Implement .csv and .exl feature

Week 21 - Write test code for .csv, .exl, and website

Week 22 - Research and Design Google Maps API implementation

Week 23 - Develop google maps API implementation

Week 24 - Write test code for google maps API implementation

Week 25 - Research training and tuning AI models

Week 26 - Train AI model on specific reviews for a specific company

Week 27 - Research and design automated process for training ML models

Week 28 - Develop and test automated process for training ML models

Estimated time before stretch goals: 42 hours

Estimate time after stretch goals: 92 hours

It doesn't require too much, and I have a point where if I get the main stuff completed, I have more things that I can do. I basically have a minimal viable product, and can make it more professional and working with more time. I've described what a finished product looks like for myself in an earlier bit of the paper, where basically I have a working backend. Once I get that finished, I can work on the extra stuff, like a frontend, maybe a payment system, etc...

Resources

I'm going to need a course or video of some kind to re-teach me how to make an API.
I'll use the PEP 8 Style guide to make sure my code is readable for most programmers.
I'll use AI as a second pair of eyes on the code I've written, as a guide to help answer questions, but I will not be copying and pasting the AI code for the project. I'll have it cite the documentation, so that I can actually learn from the source. I'll also use AI as a guide to what other resources I can rely on.
Opal, to limit phone use. Time Clock to keep track of time, with breaks within the time frame so I don't get burned out.
None of the above should cost anything, but hosting my product will cost money. A service like GCP only costs per usage, and I don't know whether or not I'll host the DB there too. It'll cost more if I do, and that's probably an unnecessary expense, but I also don't know another way to do it. More research will be needed in this area.
PS. There are free tiers of GCP that I can use to host a cloud run product, so, at least for the semester, it should be free. Might use Firebase, still researching.

Dependencies

Python: Flask, Torch, Transformers
Postman
Ollama, Chat GPT, Claude 3.5 (For reference and teaching)
GCP
Docker (Requires WSL)
VS Code
Windows, and servers
Hugging Face open source models, maybe hosting
I shouldn't need permission or licenses or anything for these dependencies. All of them are open source.
I'll be developing this on my local machine, and deploying it on GCP (I just think GCP is specific enough that'll be ok)

Risks

With API generation and money, there's always a risk, so part of that end of the project will involve pricing risks. If I have a DB hosted, and I just let it run all day and all night, with the listening component on, it will cost way too much. So part of the project will be figuring that all out.

Time would be the other factor as part of this. I will spend 126 hours or more on this project, but I don't know how much time building an API will take, how long

the hosting will take, and how long testing will take. But! I’ve got 2 semesters to figure it out and to really work hard.