# Exploring Machine Learning and Sentiment Analysis

*2024 MYP Personal Project Report*

Jerry (Jiarui) Hu

January 28, 2024

IB MYP Personal Project

*Overview*

# Exploring Machine Learning and Sentiment Analysis

Jerry Hu
2024 MYP Personal Project Report
January 28, 2024

## *Planning*

### My Learning Goal.

I didn't need to think about brainstorming ideas and topics for my Personal Project like others did—I was certain that mine had to be related to *Computer Science* and trained *Machine Learning networks*. This has been my interest and dream career since I was in 7th grade. I have spent a lot of time on coding and have accomplished numerous projects both with a team and by myself. I wanted to create a machine learning model that would recognize where my hand is, move the cursor, and perform clicks/drags using only my hand. I will secure the camera to the top of my head using a helmet, as inspired by the Apple Vision Pro. These were my initial thoughts for my Personal Project. I even researched relevant information about different algorithms that can be used to optimize the program and types of neural networks commonly used in deep learning:

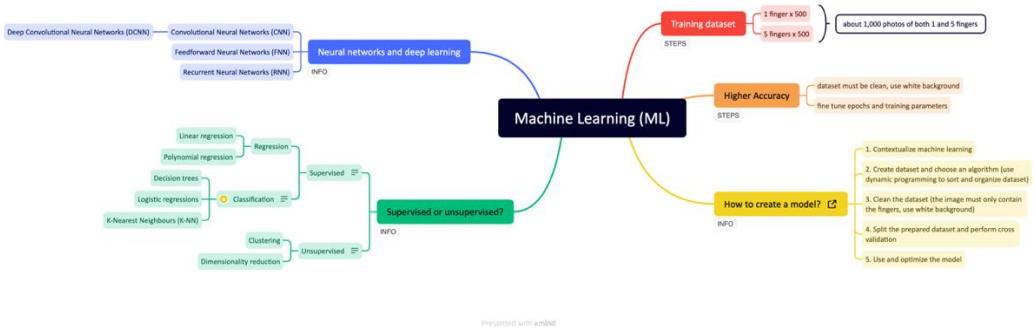



*Fig 1. Mind map of hand tracker*

However, I quickly realized that this idea would not work due to the technical limitations of my Mac. I had a 2019 Intel Mac, which does not have the capabilities to process each frame of the live feed and analyze my hand before maneuvering the cursor movement in a split second. This led me to another project that I always wanted to create—an app that would enhance a user's productivity using AI (*Artificial Intelligence*).

I wanted to create a machine learning model where the user can upload audio files to the program, and the model would be able to transcribe the uploaded audio using OpenAI's Whisper model, and generate relevant text based on its content. Through this project, my goal is to gain a deeper understanding of AI models and their interaction with users, and to apply these skills to a real-world project. I've done numerous projects of connecting APIs/apps to users; thus, I want to use this opportunity to push myself further on this pathway, exploring the use of machine learning in *NLP (natural language processing)* and *sentiment analysis* to create notes of recordings from scratch. I wanted to learn about *sentiment analysis* and how OpenAI's Whisper model can recognize different accents and languages, delving into the science behind trained *machine learning networks*. I made a new mind map for this program, called *GPTNotes*:

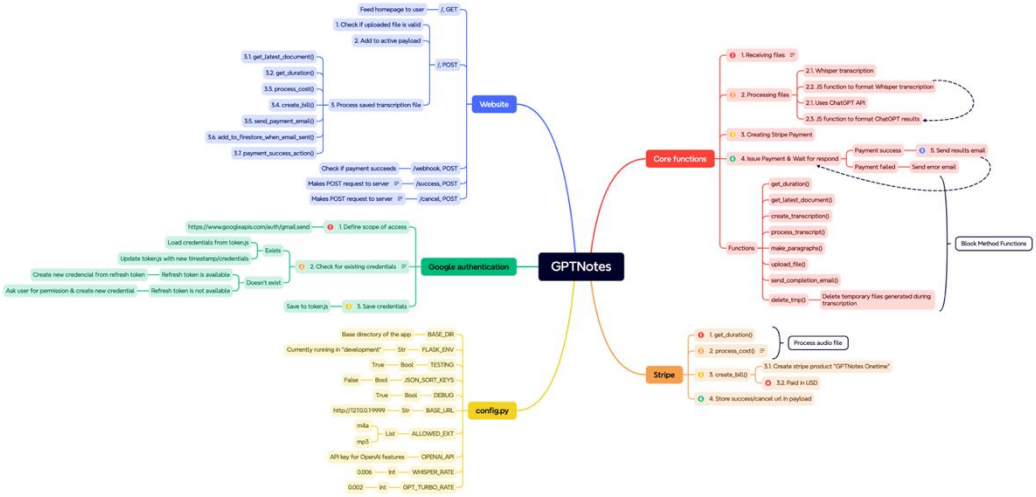



*Fig 2. Mind map of GPTNotes*

**My Product Goal.**

I am a slow note-taker. Not being able to take notes fast enough to keep up with the teacher is one of my most crucial problems in school. My goal for this product is to be able to process audio files that the user uploads and, using *trained neural networks*, convert them into detailed notes with information about the topic that is elaborated in the audio file. My hope is that *GPTNotes* will facilitate my independent exploration in the field of *machine learning* and *CNN* (*Convolutional Neural Networks*) classifier algorithms. *GPTNotes* should integrate cutting-edge speech recognition technologies with intelligent algorithms that I learn over this period of time to automatically transcribe audio recordings, PowerPoint presentations, and generates potential action items based on the context.

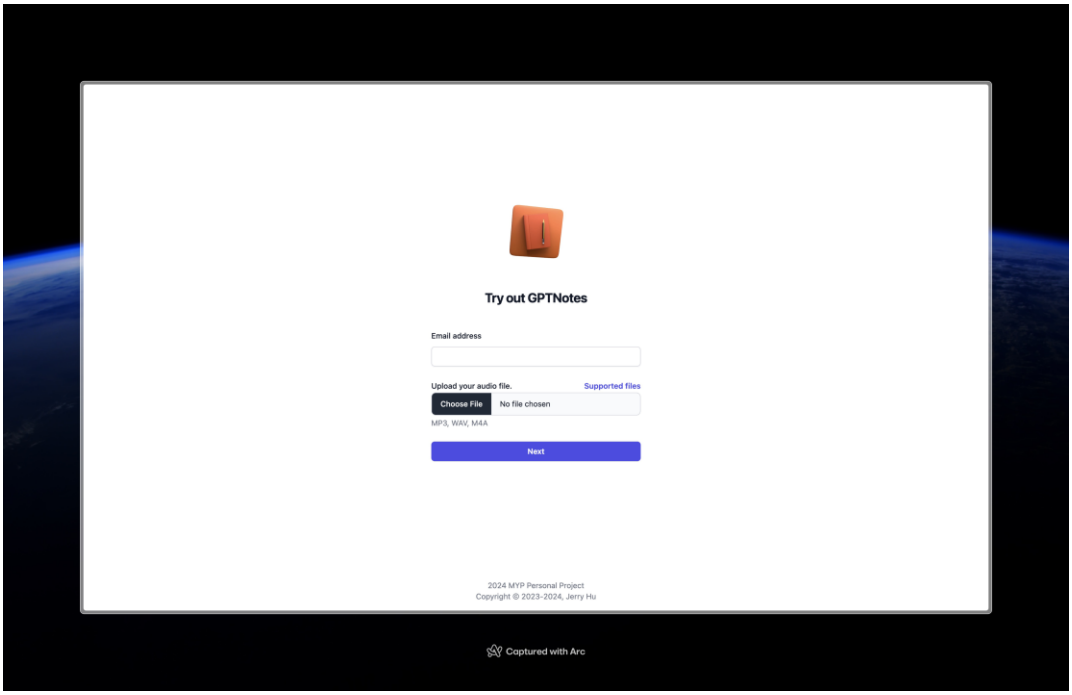



*Fig 3. Screenshot of GPTNotes' Website*

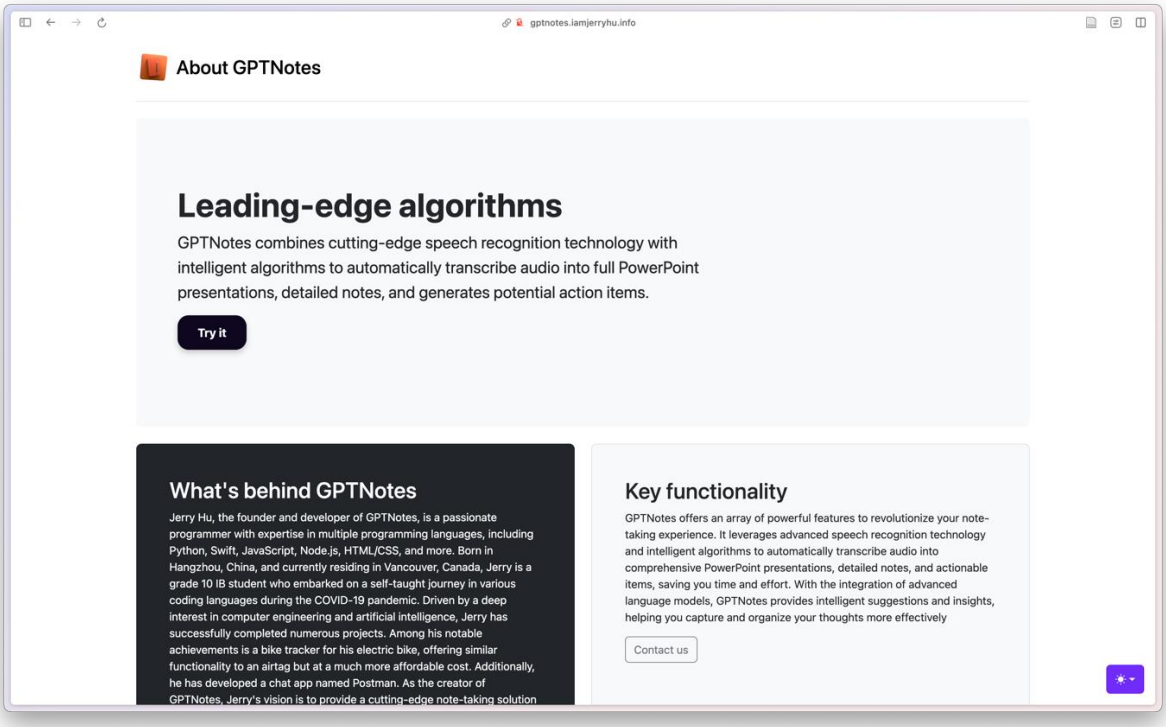



*Fig 4. Screenshot of GPTNotes' Promotion Website (About page)*

**Success Criteria.**

| **Success Criteria** | **Description** |
|---|---|
| *Accuracy/Function*<br>• A high-accuracy model is the first step towards success. Users want fast and high-quality transcriptions.<br>• Since the purpose of the IB Personal Project is to explore and learn in the area that we are interested in, I want to spend time digging into the roots of utilizing AI models to transcribe AI recordings.<br>• Given that this is the first time that I have ever done this type of projects, I will consider it successful if the app is able to transcribe two minutes of recording in less than one minute. | *GPTNotes* aims to be a speech-to-text application with built-in AI-enhancing features, such as real-time chat (to be implemented in the future), to help users capture their thoughts and ideas quickly and easily, delivering high-quality transcripts more efficiently. I've always been interested in making technology more accessible and connected to our daily lives. I will focus on achieving an accuracy of at least 95%, transcribing two minutes of audio recording in under one minute, and handling different audio qualities, accents, and background noises. This accuracy can be considered as trustworthy as having someone transcribe the recording manually. This percentage can be calculated by comparing the original audio transcript to the transcript produced by *GPTNotes* and calculating the correctly transcribed words over the total number of words. User feedback will be taken throughout the development process to ensure that *GPTNotes* meets the users' needs and expectations. I will make sure that the users' recordings and transcript meta files are |

| | |
|---|---|
| | deleted after each transcription. I will use *Google's Firebase*[1] cloud server to remotely manage the app and use *Firebase's Firestore* database to store the transcription and app data. |
| *Code Organization*<br>• I will consider this successful if all major parts of the code are commented and are formatted correctly.<br>• I will code in Python but use block-coding methods for better organization. | In order to make my code easy for others to understand, I will be using clear and consistent formatting by adding comments throughout major parts of the code. I want to use block-coding methods to help me organize the different parts of my code visually. This makes it easier for me and others to understand how the code works and make changes to it in the future. |
| *Error Handling*<br>• The console on both the terminal and the user-side browser console logs should be well-formatted and organized well, even when unexpected results are encountered.<br>• I will log necessary parts of the code for debugging, which can help me identify the parts that are working, and which are not). | I will be setting up clear and informative logging for both the console output (terminal) and the user-side browser console. I will log key parts of the application that can provide insights into what is working and what needs my attention for debugging. This will make my life easier when debugging, helping me pinpoint problems in the code quickly. |
| *User Experience*<br>• An intuitive and easy-to-learn user interface, leading to a positive user experience → more users → a more successful app.<br>• A website to promote the app. | I want to focus on having an intuitive design combined with smooth animations in the user interface. I will create a web interface that's easy to learn and understand so that anyone can use it. By making the app enjoyable and efficient to use, this will encourage more people to try it out and keep coming back to use it. Additionally, having a dedicated website to promote my app can be a great way to reach a wider audience and generate interest in my application, *GPTNotes*. |
| *Efficiency*<br>• The code will not take hostage of unnecessary CPU/GPU usage, and it should be able to maintain its performance without dramatically draining the battery life or performance of my Mac. | In order to create a smooth and sustainable experience for the users, I will make sure that *GPTNotes* avoids overburdening my Mac's resources because the server is currently running on my Mac. This means keeping the CPU and GPU usage reasonable, ensuring the server does not significantly impact my Mac's battery life, and maintaining overall performance without bogging down the system. After conducting thorough research on a typical app's memory impact on a |

[1] "Firebase is an app development platform that helps you build and grow apps and games users love. Backed by Google and trusted by millions of businesses around the world." —*Official description of Firebase.*

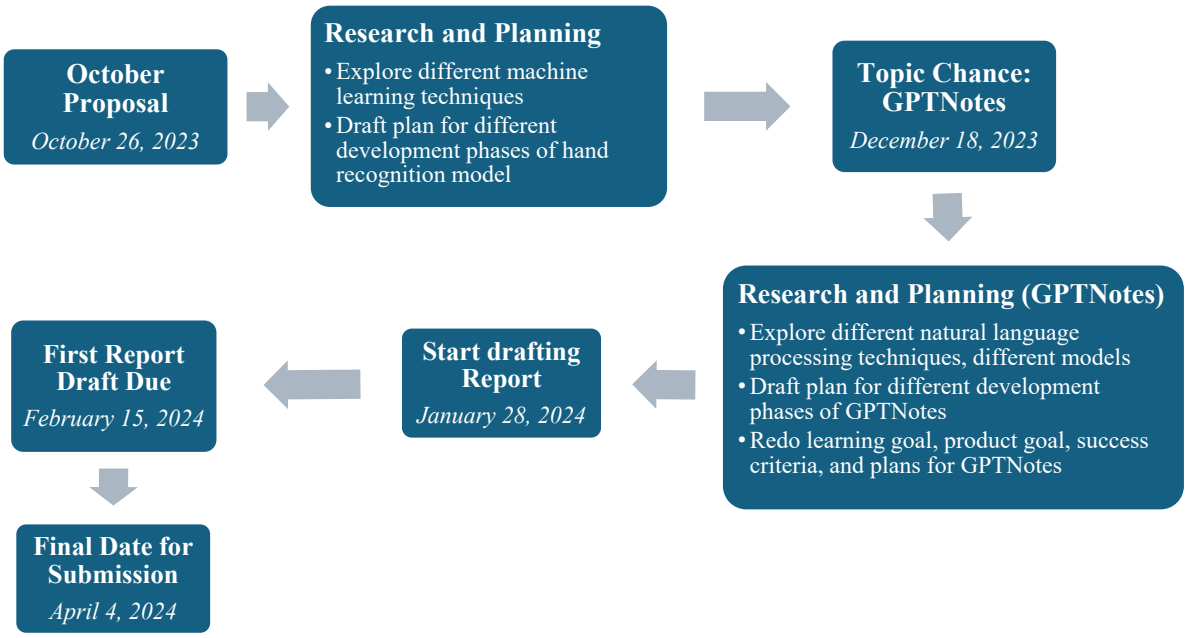
| | |
|---|---|
| | typical 2019 Intel Mac, I have narrowed down my target efficiency to less than *17%*. *GPTNotes* should be able to process the recording of any length using only *17%* of ram. A typical on-device *machine learning* app operates using about 15-20% of memory. |
| *Aesthetics*<br>• Aesthetics is an important aspect in apps because it's linked to various criteria: functions, organizations, user experience, etc.<br>• The app (including terminal/console log page) should be outputted in a neat and organized way, so that users can correctly interpret what's going on with the app, and what each error means when one is produced (which also falls into the error handling success criteria category).<br>• The website will be modern looking, efficient, and straight to point. | I want to create an accurate and efficient application, with clear code organization and, informative error handling. Additionally, I want to create a smooth animation and having an intuitive and simple interface that feels natural to use, avoids unnecessary complexity, and leads the user to the core features. I believe that aesthetics is the key to attracting users. |

**Timeline.**
This table below focuses on the timeline of my personal project and the important deadlines that my product needs to have. It functions as a reminder for what I should work on each week and the deadlines that are due at the end of each month.

| **Deadline** | **Tasks** | **Descriptions** | **Relevant Success Criteria for the task** | **Notes** |
|---|---|---|---|---|
| October 26, 2023 | Finish October Proposal | The October Proposal is due on October 26, 2023. | N/A. | Completed before time. |
| November 30, 2023 | Finish Research and Planning | Explore the different machine learning techniques, draft a plan for different development phases of the hand recognition model. | N/A. ATL Skills: Researching and self-management skills. | I looked into a few free courses on Coursera to learn different *ML* techniques and which ones to use for the hand recognition model. |

| | | | | |
|---|---|---|---|---|
| Early December (December 3, 2023) | Finish research and start with first prototype. | I used OpenCV and Python to create the first prototype. To train the program to recognize my hand, I used the *hand haarcascade.* | Code organization, Accuracy, Function, Error handling. | I took four hundred photos of my two hands with a white background and four more hundred photos of my two hands with a random background. |
| December 18, 2023 | Topic Change: *GPTNotes* | Topic change to *GPTNotes* is approved. The low-FPS and lag issues on the original model would make the app practically unusable. | N/A. | I changed my topic because the FPS issue I was trying to solve was a technical limitation of my Mac, which is not something I could solve. |
| End of December (December 27, 2023) | Train and test machine learning v2 model. | The model should be trained and tested by Mid-December, so that it can be put into the program afterwards. | Code organization, Accuracy, Function, Error handling. | I trained the model *epochs* using the eight hundred pictures I took of my hand. Wrote the model code in block-method so that the code is organized. |
| December 31, 2023 | Research and Planning | Explore different natural language processing techniques, different models, draft a plan for development phases of *GPTNotes*, redo the learning goal, product goal, success criteria, and plans for *GPTNotes*. | N/A. ATL Skills: Researching and self-management skills. | Completed during winter break. |
| Mid-January (January 14, 2024) | Finish prototype | The first draft/beta of the program should be functioning at this stage. | Code organization, Accuracy, Function, Error handling. | Completed on time. I published the final working version of *GPTNotes* onto GitHub. |
| January 28, 2024 | Start writing the final report | Work on the final report. Started on January 28, 2024. | Accuracy/Function, Efficiency | Started on time. |

| | | | | |
|---|---|---|---|---|
| Early February (February 1, 2024) | Finish product | Between mid-January and Early February is where the product will be tested and adjusted for bugs and changes. The product should be completed by Early February. | N/A. ATL Skills: Time management, self-management. | Completed on time. I will also start the first draft of my report after I finish my product. |
| February 15, 2024 | First Report Draft due | The first draft of my report is due on February 15, 2024. I should review my report with my supervisor and determine changes that needs to be fixed in both my product and report. | Accuracy/Function, Efficiency, ATL Skills: Self-management, communication skills. | Completed on February 14, 2024. I effectively demonstrated my self-management skills in that I finished my final report earlier than the expectation. |
| April 4, 2024 | Final Date for Submission | Last day of Personal Project submission is on April 4, 2024. | ATL Skills: Self-management and communication skills. | After communicating with my supervisor (I will be in China on the first week of April and ManageBac is blocked there), we have concluded that I will try my best to finish and hand in my personal project report before I leave. This will be a challenge to my self-management and time-management skills. |
| May 1, 2024 | Ten Talks | Ten Talks happens on the afternoon and night of May 1, 2024. | N/A. | N/A. |




**Action Plan.**

1. Determine the topic.
2. Come up with a plan using a mind map.
3. Research how the *machine learning* models can be created and trained.
4. Read OpenAI's documentations.
5. Create the prototype of the app.
6. Create a website displaying the app. I will add a connection between the website and my server in the future so that users can try *GPTNotes* directly from the website. This is to be completed in the future.
7. Write the server code (the real program code and the training code for the model).
8. Connect the app (server code) to OpenAI's models, preferably Whisper.
9. Develop the model.
10. Train the model using large training data (using different writing samples, etc.).
11. Test the model.
12. Incorporate the model into *GPTNotes*.
13. Test *GPTNotes*.
14. Start the first draft of the report.

## *Applying Skills*

**ATL Skills.**

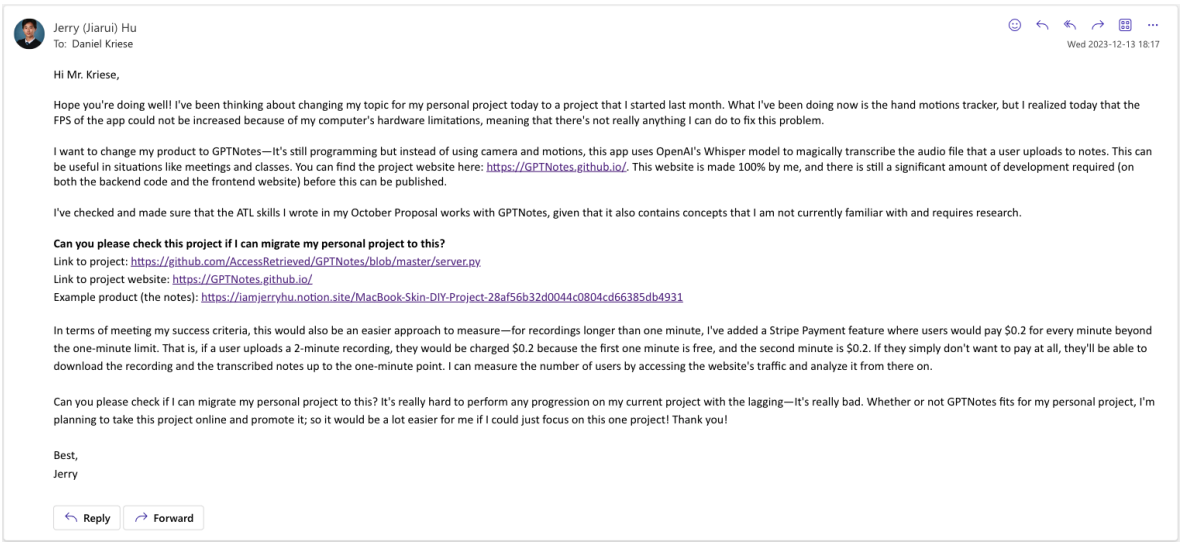
Just to remind myself, here are the SHAPE skills that I set in my October Proposal:

*Researching Skills*—To enhance *GPTNotes*, I will research *NLP (Natural Language Processing)* and audio transcription algorithms. Looking into OpenAI's Whisper model and other machine learning techniques for analyzing audio will be crucial.

*Communication Skills*—For connecting with my supervisor, commenting code (markup different sections of code so it is easier to interpret), reading articles and websites on *CNN* (*Convolutional Neural Network*), papers, and forums for helping with my code such as Stack Overflow or Stack Exchange.

*Self-Management Skills*—Given that it is a long-term project, being able to manage myself (including time management) to effectively manage my time and project milestones would be crucial, making sure I don't procrastinate over time.

While working on *GPTNotes*, I actively utilized my research skills to explore the realms of *natural language processing* and audio transcription algorithms. This exploration and learning process was pivotal in establishing the technical foundation of the server-side program (which took me the longest). I used Stack Overflow and Stack Exchange whenever I encountered bugs and problems that were beyond my ability. Communicating effectively played a key role in my learning too; whether it was in discussion progress with my supervisor, ensuring my code was well-documented for readability or engaging with resources and communities online to deepen my understanding of *CNN* (*Convolutional Neural Networks*). This became especially useful when I decided to change my Personal Project topic. I diligently managed my time and adhered to the project checkpoints I set earlier, which was indispensable for both my successful learning and the development of *GPTNotes*.



Jerry (Jiarui) Hu
To: Daniel Kriese

Wed 2023-12-13 18:17

Hi Mr. Kriese,

Hope you're doing well! I've been thinking about changing my topic for my personal project today to a project that I started last month. What I've been doing now is the hand motions tracker, but I realized today that the FPS of the app could not be increased because of my computer's hardware limitations, meaning that there's not really anything I can do to fix this problem.

I want to change my product to GPTNotes—It's still programming but instead of using camera and motions, this app uses OpenAI's Whisper model to magically transcribe the audio file that a user uploads to notes. This can be useful in situations like meetings and classes. You can find the project website here: https://GPTNotes.github.io/. This website is made 100% by me, and there is still a significant amount of development required (on both the backend code and the frontend website) before this can be published.

I've checked and made sure that the ATL skills I wrote in my October Proposal works with GPTNotes, given that it also contains concepts that I am not currently familiar with and requires research.

**Can you please check this project if I can migrate my personal project to this?**
Link to project: https://github.com/AccessRetrieved/GPTNotes/blob/master/server.py
Link to project website: https://GPTNotes.github.io/
Example product (the notes): https://iamjerryhu.notion.site/MacBook-Skin-DIY-Project-28af56b32d0044c0804cd66385db4931

In terms of meeting my success criteria, this would also be an easier approach to measure—for recordings longer than one minute, I've added a Stripe Payment feature where users would pay $0.2 for every minute beyond the one-minute limit. That is, if a user uploads a 2-minute recording, they would be charged $0.2 because the first one minute is free, and the second minute is $0.2. If they simply don't want to pay at all, they'll be able to download the recording and the transcribed notes up to the one-minute point. I can measure the number of users by accessing the website's traffic and analyze it from there on.

Can you please check if I can migrate my personal project to this? It's really hard to perform any progression on my current project with the lagging—It's really bad. Whether or not GPTNotes fits for my personal project, I'm planning to take this project online and promote it; so it would be a lot easier for me if I could just focus on this one project! Thank you!

Best,
Jerry

Reply Forward

*Fig 5. Screenshot evidence of my email to supervisor*

**Resources.**

*What is one big question that you aim to answer?*

How can the use of advanced machine learning models, like OpenAI's Whisper or Google's Gemini Ultra/Advance, affect the quality of transcription and analysis of audio recordings in real-time? If this were to be used in a meeting or a lecture, how can this improve the meeting/classroom qualities?

*Response.*

After working on *GPTNotes* and getting in touch with all the different types of machine learning models, I learned that the Whisper and Google's Gemini Ultra are similar, but not the same. Whisper is a machine learning model made for speech recognition and transcription only, while Gemini Ultra functions as a *large language model (LLM)*, serving as the replacement/successor to *LaMDA*[2] and *PaLM 2*[3]. If applied to a meeting or lecture setting, these advanced *machine learning models* can improve people's experience by providing an instant, accurate text versions of the spoken content, which can be beneficial for people who may have difficulties following along due to language barriers or hearing impairments. Additionally, *sentiment analysis* can gauge the mood and engagement of the speaker and the audience, allowing for real-time adjustments of the transcript. This can lead to a more interactive and responsive user experience.

*List the primary and secondary resources you will need to consult to complete your research.*
Primary—OpenAI's API documentations, GPT models, Whisper model, Stack Overflow/Stack Exchange, and GitHub.
Secondary—Reports, academic papers on machine learning, community forums, existing code repository on GitHub.

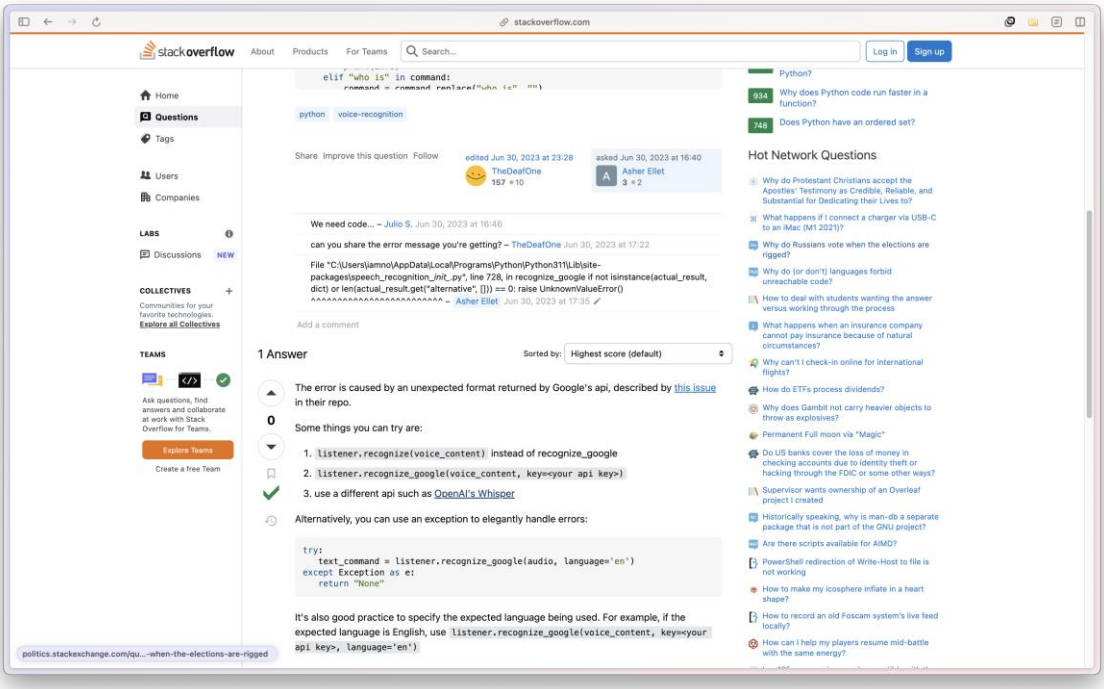



*Fig 6. Screenshot example of a Stack Overflow response that helped me.*

**Possible Inquiry Questions.**

1. What is machine learning? How does it work?
2. How can machine learning and generative AI help with our daily lives?
3. What is a sentiment? What is sentiment analysis?

[2] *LaMDA:* Language model for Dialogue Applications.
[3] *PaLM 2:* Path-augmented Language Model 2.

4. How does OpenAI's whisper model work? Why can OpenAI's Whisper model recognize different accents and languages?
5. What is deep learning?
6. What is the difference of deep learning compared to machine learning?
7. How to train a machine learning model?
8. What is the difference of AI and machine learning?
9. How will researching advanced algorithms like OpenAI's Whisper model influence the develop of *GPTNotes*? How did it change *GPTNotes'* technical foundation?
10. In what ways will communicating with my supervisor and engaging in coding forums like Stack Overflow contribute to my learning process?
11. How will I plan and manage my time effectively to ensure steady progress on my Personal Project?
12. What challenges did I encounter while coding *GPTNotes*? How did I apply critical thinking skills to overcome them?
13. How will I evaluate and select the most relevant information from the vast resources available on the internet about natural language processing and machine learning? How to ensure that they are credible?
14. How does my project align with my interests in computer engineering and AI?
15. Reflecting on the entire project, what would I do differently if I had the chance to start over?

**Achieving My Learning Goal.**

My journey through learning *machine learning* and *sentiment analysis* was full of growth and exploration. I was set on creating a real-time hand tracker that could control my Mac but had to shift to developing *GPTNotes* due to insufficient computing power (the subsequent result of having an Intel Mac, instead of the M-series chip) on my Mac to increase the FPS of the hand tracker. This change required me to invest more time into research and to delve more deeply into the arts of *natural language processing* and audio transcription algorithms. My goal was to gain a comprehensive understanding of *trained neural network models* and their interactions with users. I aimed to push my boundaries in exploring machine learning and sentiment analysis[4]. This journey not only enhanced my technical skills in terms of training neural networks, but also broadened my understanding on how AI applications work in real-world scenarios. *Sentiment analysis* is used in *GPTNotes* where the application would suggest related topics and follow-up checkboxes.

**CRAAP Evaluation of Sources.**

Source URL: https://platform.openai.com/docs/guides/speech-to-text

*Currency:* The official documentation for the *OpenAI Whisper Model* is up to date, for its last update shown on the website was in October 2023, which is recent enough given that I started my Personal Project in Mid-November. All links in this website are functional.

[4] *Sentiment analysis: "field within natural language processing (NLP) that involves analyzing text to determine the sentiment or emotional tones behind it. It is widely used in business and social media monitoring to understand what the customer wants (i.e. their reaction to a specific product)."*

*Relevance:* This documentation directly relates to my topic—Exploring Machine Learning and Sentiment Analysis. It explains how the Whisper model works and how to use it for *speech-to-text* (speech recognition) tasks.

*Authority:* This documentation is published on the official OpenAI website, which suggests that its credibility can be trusted for it's a reputable organization in the field of artificial intelligence. On their "About" page, they have included contact information such as email, geolocation, phone, fax, proving that this organization is legitimate.

*Accuracy:* The information is accurate and supported by technical details and reference because I followed closely to its guide on utilizing the model, what parameters to include in, etc. On the other hand, *GPTNotes* is working, which indicates that the information on the website is accurate.

*Purpose:* The purpose of this documentary is to inform and teach users about the Whisper model, and how to use it. It's presented as a factual and educational content, aiming to provide users like me with the knowledge they need to use the model.

**Applying Skills to Create a Product.**

*How does OpenAI's Whisper model work? How does it process audio input and convert them to text? Besides the text-processing model I created, what other options are out there that's more efficient and widely incorporated?*

The OpenAI's Whisper model works by processing audio input (this can be a live stream or a recorded file) and converts it to text. Firstly, it preprocesses the audio to normalize the *gains* and clean it. Then, it extracts key features relevant to speech from the audio signal. Using *deep neural networks* (*deep learning*), Whisper interprets these features to understand speech patterns, recognizing a wide range of language and accents because it is trained on diverse datasets. The model then uses this recognized speech patterns to transcribe the speech into text, with post-processing steps to ensure accuracy and readability. This process highlights Whisper's advanced capabilities in speech recognition and transcription.

Besides the text-processing and sentiment analysis model I created, other models such as the *DistilBERT*[5] (a pre-trained model from Hugging Face), can be used for *sentiment analysis*. Hugging Face is a platform where the "machine learning community collaborates on models, datasets, and applications," as stated on their official Home page. The *DistilBERT* model is known for its efficiency and performance in natural language data. This model is widely used in the *NLP* [6]community for its impressive balance between performance and efficiency, making it a great alternative for sentiment analysis task. However, because the Personal Project product has to be my own work, I decided to use my own text-processing *sentiment analysis* model that I created and trained for *GPTNotes*.

## *Reflection*

**Impact of *GPTNotes* on myself and my learning.**

[5] *DistilBERT:* Distilled Bidirectional Encoder Representations from Transformers. DistilBERT is a lighter, faster version of *Bidirectional Encoder Representations from Transformers (BERT)* that remains most of its predecessor's features.

[6] *NLP:* Natural Language Processing.

During the three years of COVID-19 lockdown, my love for coding went from a spark to a bonfire. Being confined in my room in the house for over a year and attended online school where it killed any social interactions I had from school; this period of isolation became a golden opportunity for self-improvement to me. From Raspberry Pi to Python Turtle, from Flask to TensorFlow, my fascination with coding was not just about understanding the syntax of different languages or reading documentations. I dedicated myself (and lots of my sleep time) to building *Constellation Networking*, a web app that assists teenagers in interacting with professionals in their future careers, and *RuiRuiJiZhang*, an accounting app that I built in Chinese. The name *RuiRuiJiZhang* derived from my Chinese nickname "RuiRui" and "JiZhang," which simply means accounting. Through these projects, I tackled challenging bugs, spent days and nights on Stack Overflow and GitHub trying to find a solution, and bringing ideas to life. SwiftUI, in particular, captivated my interest early on due to the prospects of running my own apps on my cellphone. I spent countless days and nights in front of my Mac screen, trying to build something out of the blue. However, this journey was not without bumps. XCode's high storage demands of over 50GB became a bottleneck to my low-storage Mac, and I was forced to uninstall it. Nevertheless, this setback did not dampen my interest in coding; instead, it redirected me to JavaScript and HTML/CSS. I started building my own website and began digging into developing front-end web applications in December 2020.

This self-taught journey through the landscape of coding languages and technologies has been more than just an educational endeavour. It taught me the core values of perseverance. During the early development stages of *RuiRuiJiZhang*, I often find myself spending about four hours of my sleep in my bed trying to find the section of code that caused the bug. It took almost my whole night's sleep but was very rewarding at the end. It also taught me the importance of continuous learning, and the sheer satisfaction that comes after simply seeing the "Build Successful" dialog. As I look back to those nights, I spent deciphering the complex code that I once knew nothing about, I see them not just as cherished memories, but also as foundational building blocks of my future in computer engineering and artificial intelligence.

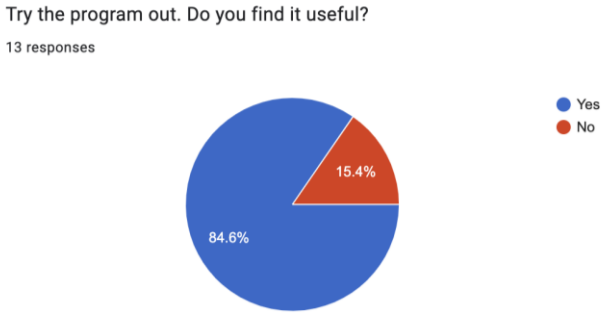
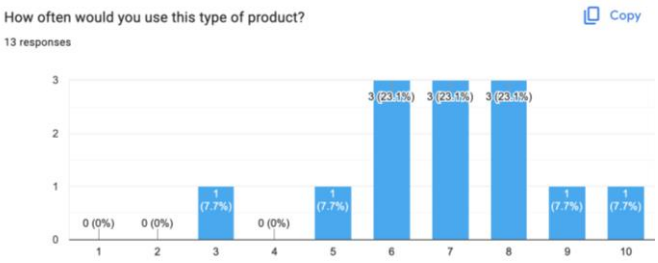
As I reminisce about those days where my Mac screen illuminated in my quiet dark room, the coding challenges that once seemed insurmountable now fuels my inner passion towards innovation. It's this passion that has culminated in the creation of my Personal Project product, *GPTNotes*—a powerful AI-powered tool. *GPTNotes* dramatically elevates productivity for me as a slow note-taker. It leverages advanced speech recognition technology and intelligent algorithms I crafted, to transcribe audio into comprehensive notes, actionable items, saving me the time and effort. Pulled back from the nostalgia of my coding adventures, I now see *GPTNotes* as a bridge between the me that knew nothing about coding and the version of myself today that's skilled in many diverse languages.

**Evaluate *GPTNotes*.**

I created a Google Forms survey that will allow different users to upload their own audio file (or use one of the audio samples) to test *GPTNotes* and give feedback. As a token of appreciation, this experiment will allow participants to use the program for free with no limitations. To evaluate my Personal Project and my product, *GPTNotes*, I have surveyed thirteen people to accurately analyze the different individuals' needs of *GPTNotes* in their daily work life. I have attached the survey results in the success criteria categories below.

| **Success Criteria** | **Evaluation** |
|---|---|

| | |
|---|---|
| *Accuracy/Function* | The final product works fairly well. It is able to process a two-minute recording in less than forty-five seconds. Because the results are being sent out by email, I did not take into account the time it takes to receive the email for that relies greatly on the network connection. In the future, I am looking to implement real-time recording in the web app, allowing users to record and process the audio without the need for another recording app or uploading to GPTNotes. However, the uploading feature will still be retained. |
| *Code Organization* |  <br>*Fig 7. Evidence of block-coding method*<br><br>Given this above image as evidence, I have coded the key parts of the code in block-method. This ensures better management and clarity. The whole program at the end has about 2,487 lines of code, covering different languages such as Python, JavaScript, CSS, HTML, and JSON. |
| *Error Handling* |  <br>*Fig 8. Evidence of error handling and logging*<br><br>As shown above, I have implemented a clear and coherent logging method, including any errors that the program received while running *GPTNotes*. One of the biggest bugs that I experienced during developing *GPTNotes* is where the *sentiment analysis* model I created sometimes returned an *empty string* because the topic wasn't included in the training datasets, leading to an email result with very limited information. I fixed this by implementing a bigger training dataset I found on Hugging Face. To better customize the training dataset, I converted the training data (file extension *.joblib*) to a *JSON file.* This conversion enabled me to modify |

| | |
|---|---|
| | the contents of the training data. I added topics and conversations that are commonly used by teenagers. |
| *User Experience* | As shown in *figure 3*, the app features an intuitive and easy-to-use interface, featuring an email input box where the user will receive the results, and an audio upload box with clear instructions of what file formats are accepted. As shown in this response, the majority (*84.6%*) tried my program and found it useful:<br>  <br><br>*Fig 9. Evidence of user experience*<br>  <br><br>*Fig 10. Evidence of user experience* |
| *Efficiency* | While executing the program, I referred to *Activity Monitor*, which is an app on my Mac that allows me to read the memory and CPU/GPU usage each app is using. It is a native Apple app[7]. *Activity Monitor* did not output any logs that indicates *GPTNotes* consuming unnecessary CPU/GPU usage or draining the battery life dramatically. According to the survey results, the majority of the people equally believed that the program had either 801-1000 lines, or 3001-5000 lines. The entire *GPTNotes* program was written in 4,487 lines of code. I would consider it a success given that *GPTNotes* is capable of running over 4,487 lines of code to process a two-minutes audio recording in just under forty-five seconds, which is efficient enough for daily-use purposes. |
| *Aesthetics* | As shown in *figure 3* and *figure 7*, the web app features a modern and easy-to-understand webpage and console log. The website uses the latest Tailwind CSS features, making it look modern looking, efficient, and straight to point. |

My Process Journal on Notion can be accessed here: https://iamjerryhu.notion.site/Process-Journal-a68a12337f6646569872d300715046ac

[7] *Native apple app:* A software application specifically designed and developed to run on Apple devices.