1. There are two parts to this Chrome extension itself that can be loaded into google chrome. It currently can change the background color of the webpage by selecting a color and then pushing the read text button. The second part involves it scraping the text of the webpage and sending it to a python server which modifies it based on links and keywords and highlights them. Sending it back and displaying it in the extension. This second part of the code will not work if the server is not running.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. This seems to be working good. Though there are two versions included. PowerpointAudio.py is what was shown to us and uses the sphinx speech recognizer which does not work that well. The better working version is PyAudioRecorder.py which uses the Google speech recognizer which is much better but requires a steady internet connection and a network that won’t block its traffic. The University network will block this from all attempts I have done.
2. With the amount of prep put into this data at the moment I believe that a random forest would be the best model to use to figure out if This is the most accurate. Since it is essentially a ensemble model of decision trees it works well with sorting through variables that don’t help much. Starting off even before I started dropping variables or adjusting the parameters of the model the decision tree was getting an accuracy percentage of over 98%. Which shows that it was able to successfully predict 98% of the test data based on the training from the training data. Neural networks also did good and less prone to negative changes to the variables that were excluded. I think overall the Neural network is the model to go to if you want to use a small number of variables to limit the amount of data needed while a decision tree can more effectively utilize a large amount of variables. With the data prep I did the Bayes classifier didn’t do as well. Don’t have as much experience with it so I’m not sure how to work with it as well.
3. So the AI I used to summarize was Copilot and Copilot professional. Copilot summarized it in a way that exaggerated the urgency of the phishing emails in a way that made it more obvious it was a phishing email. Though if you don’t have much experience then the way it did it might put more stress and could lead to some people becoming more likely to fall for these. Copilot professional on the other hand outright detected that these might be phishing emails and included notes in its summarization to be careful and provided mitigation steps to reduce the likelihood of someone falling for it.
4. I think ultimately, I would want to have my emails summarized by an AI that has been trained in some part to detect phishing emails. Looking at the results I believe CoPilot professional might have had that be a part of its training. My Python summarization programs mostly just extracted important details from the email or got rid of unneeded information. They are not able to infer that much without the implementation of machine learning. One Method I did investigate is possible using an api with Chatgpt to submit custom prompts to get to summarize the email and to check and see if the email was a phishing email. Ultimately, I paused that as I needed to purchase tokens to continue this but I think that would be the route to go for summarizing emails.