PROGRESS REPORT (PROJECT DIARY):

Name: Neilabh Banzal

Project: Unscripted

DATE	PROGRESS
Before 12.04.21	Read • https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471 • https://medium.com/@sanchittanwar75/introduction-to-machine-learning-and-deep-learning-bd25b792e488 • https://www.youtube.com/watch?v=aircAruvnKk Comfortable with • NumPy • Matplotlib • Pandas • Jupyter Notebooks Watched Git Session
12.04.21 - 19.04.21	Read • Hands-On Machine Learning (Ch 1-4) • Python Machine Learning, 3rd Ed. (Ch 1-4)
19.04.21 - 15.05.21	No updates due to exams and project submissions.
15.05.21 - 22.05.21	Read • Examples of Huggingface • Examples of Torch Audio For Text Summarisation Read https://pypi.org/project/bert-extractive-summarizer/, but did not implement it. Implementation on To Do.
23.05.21 - 30.05.21	Read https://github.com/manan-paneri-99/n2p-gpt3-bot • Documentation • Went through the code Finalised the library I want to try out - SpeechRecognition Library https://pypi.org/project/SpeechRecognition/3.8.1/ https://github.com/Uberi/speech_recognition#readme
05.06.21 - 11.06.21	Read further on STT - • https://heartbeat.fritz.ai/a-2019-guide-for-automatic-speech-recognition-f1e1129a141c • https://realpython.com/python-speech-recognition/ • https://www.analyticsvidhya.com/blog/2021/01/introduction-to-automatic-speech-recognition-and-natural-language-processing/ • https://medium.com/georgian-impact-blog/how-to-make-an-e

	nd-to-end-automatic-speech-recognition-system-with-wav2vec-2-0-dca6f8759920 Implementation • Used Google Speech Web API using the SpeechRecognition Library on the preset audio file to get an estimate of WER • Details - https://drive.google.com/drive/u/0/folders/1_UAroBYlWUnVwTRDl22_e4BX0L5WdwPg • Further Details and implementation - https://github.com/Neilabh21/SoC21-Unscripted/blob/main/SoC_Unscripted_TTS.ipynb
13.06.21 - 20.06.21	Reading - • Precision and recall: https://en.wikipedia.org/wiki/Precision_and_recall • How to evaluate STT models: https://medium.com/ibm-data-ai/how-to-properly-evaluate-speech-to-text-engines-c90fa902667f • WER calculation: https://aws.amazon.com/blogs/machine-learning/evaluating-an-automatic-speech-recognition-service/ • N-grams: https://en.wikipedia.org/wiki/N-gram • Bleu vs Rouge: https://stackoverflow.com/questions/38045290/text-summarization-evaluation-bleu-vs-rouge • BLEU: https://en.wikipedia.org/wiki/BLEU • ROUGE: https://towardsdatascience.com/the-ultimate-performance-metric-in-nlp-111df6c64460 Implementation • Extended the implementation using Google Speech Web API to 5 audio files, and performed analysis using WER, ROUGE, BLEU metrics • Read https://www.journaldev.com/46659/bleu-score-in-python to finalise the arguments for ROUGE and BLEU metrics. • Pushed updated code to GitHub - https://github.com/Neilabh21/SoC21-Unscripted/blob/main/SoC_Unscripted_TTS_2.ipynb
20.06.21 - 23.06.21	• Survey for deciding the final project - https://docs.google.com/document/d/1HtRJGp6ldfmAEjRkhJu8C_yNOGXJv-fFzus1Got42ww/edit?usp=sharing
23.06.21 - 27.06.21	• Literature Survey for Meeting Minutes from audio files. (5 hours) • Finalised flowchart (1 hour)

27.06.21 - 04.07.21	• Discussed and finalised the plan of action (2 hours) • Less work due to unavailability
04.07.21 - 11.07.21	• Explored GPT-3 based abstraction models • Not available for the public • Applied for a research account (Total - 3 hours) • Tried out different Abstractive Summarisation Codes (~5) • The only one that worked as desired is here. (Total - 5 hours) • Looked at 2 alternatives - haven't tried out yet. • https://pypi.org/project/pysummarization/ • https://www.thepythoncode.com/article/text-summarization-using-huggingface-transformers-python • Set up the Chrome Extension to record Google Meet for generating data for validation.
12.07.21 - 18.07.21	• Convert the current research into a pipeline for Converting Meet Transcripts to Summaries. (8 hours) • Needed regex to remove unnecessary portions. (2 hours)