COMP 5600 Project Proposal Report
Laura Wilson (lgw0020@auburn.edu)
Coordinator: Daniel Harrison(dah0052@auburn.edu)
Taylor Cross(tac0062@auburn.edu)

- Define clearly what dataset you will work on. (What is the primary application domain? What are the inputs? What are the outputs?)

Our dataset will take different audio files and classify them into different speech commands. This could be applied to robots and with large language models. One could use this to give commands to a robot that they can interpret in more complex tasks. More specifically keyword classification. Be able to take audio and hone in on keywords.

- How Machine Learning techniques can be applied to solve this problem?
  We will use multiperceptions to solve the machine-learning tasks.

- What are the potential challenges you may face in this project?
  The dataset is complex, has many moving parts, and will be computationally expensive. Also we will need to adapt the dataset for use in a MLP.

- What is your dataset? How will you create/build your dataset?
  Our dataset is audio speech commands. It contains words used to train speech recognition. It also contains audio files of silence and static to help differentiate them from words.

- How do you plan to build it? Identify what technologies you plan on leveraging to implement your software. This may be programming languages, supporting libraries, etc.
  The dataset we are using is already built and being used. Our environment will need libraries like TensorFlow or PyTorch, Librosa for audio processing.

- What hyper-parameters will be involved in your classifier? How would you fine-tune these hyper-parameters? (optional for 5600s)
  We will need to use hyperparameters to fine-tune our data and make it work for our purposes. We will need several hypervisors including using optimization algorithms, and learning rates. The data comes with a split dataset with validation functions on them.

- How will you demonstrate the usefulness and correctness of your classifier?
  We will demonstrate the usefulness by making the model detect when a single word is spoken. Correctness will be demonstrated by comparing the model's answer to the correct answer in the dataset.

- Provides a rough timeline to show when you expect to finish what. List a couple of milestones if possible (they can be tentative).

March 29th - 31st: Environment Setup

Set up our development environment, including necessary libraries (e.g.,

TensorFlow or PyTorch, Librosa for audio processing).

Install Renumics Spotlight and other tools for data exploration.

March 31st - April 2: Data Exploration and Preprocessing

Load the SpeechCommands dataset and explore it using Renumics

Spotlight to understand its structure and contents.

Begin feature extraction (MFCCs, spectrograms) and normalization.

Handle variable audio lengths through padding or truncation.

April 2 - 9: Model Development and Initial Training and Model Architecture and Baseline Training

Design an initial MLP model architecture. Start with a simpler model to establish a baseline.

Train the model on our preprocessed and augmented dataset. Evaluate baseline performance on the validation set.

April 9 - 11 Hyperparameter Experimentation

Experiment with basic hyperparameters (e.g., layer sizes, learning rate, activation functions).

April 11 - 16: Evaluation and Refinement

Produce Progress report

Begin refining the model based on initial results, possibly revisiting data preprocessing and augmentation strategies.

April 16 - 18 Advanced Hyperparameter Tuning

Dive deeper into hyperparameter tuning, exploring regularization techniques, dropout rates, and optimization algorithms.

Consider using automated hyperparameter tuning tools like Hyperopt or Optuna if available.

April 18 - April 20: Extended Training and Evaluation

Train the model with our refined hyperparameters and architecture.

Perform extensive evaluation using the validation set.

Start analyzing model errors to identify potential areas of improvement in data preprocessing, feature extraction, or model architecture.

April 20 - 22: Incorporation of Enrichments and Metadata

Integrate dataset enrichments (e.g., label error scores) into the training process, potentially for filtering or sample weighting.

Conduct further experiments based on insights gained from the dataset's metadata and enrichments.

April 25- April 26: Documentation and Reporting

Document the model architecture, training process, hyperparameter settings, and performance metrics.

Prepare a report or presentation summarizing our findings, challenges encountered, and areas for future work.

Review our project to ensure all components are complete and well-documented.

Submit our project, code, and any deliverables as required.