Bioacustics initial notes:

* The main problem is separating background noise from the signal noise we want to capture (the whale calls initially, but we may look into other items later).
* We can focus on visual detection w/ spectrograms. Look for patterns in the images that correspond to known whale calls. (We could take a kind of non-machine learning approach to this, instead trying to just literally find near identical matches to known calls. This would depend on the similarity/variance of known calls. We could just do feature detection in general, try and identify and signal that seems unlikely to have come from background noise. That could be less prone to overfitting errors)
* We could also focus on raw audio data detection (SINCNET), although the spectrogram contains identical data to the raw audio data
* Has any baseline testing been conducted on pervious teams models?
* We want to use a sampling rate twice as high as the highest frequency we want to capture
* Use multiple metrics to obtain a holistic picture of how good the model is (ROC, PR curve, DET curve, \*\*cost curve\*\*)
* Marine mammal calls can be complex and variable (Allen et al.,2018) and evolve seasonally or annually (McDonald et al., 2009)
* neither is the noise encountered in the ocean strictly stochastic (Livina et al., 2018) (Can we actually model the background noise itself and clean it out of the images before attempting classification/detection??)
* Some of the factors that lead to this variability are related to the animals that produce these sounds including: changes in the source level, changes due to behavioral state, geographic variations in sound production, animals’ demographic differences (e.g., age and sex), and even orientation of the animal with respect to the sensor (particularly for high frequency echolocation clicks)
* There exists highly accurate CNN for classifying individual segments of sound as containing particular calls, although it seems like for the most part these are generally just binary classifiers. If we want bounding boxes we could try and use feature extraction on the CNN to see which segments of a image are causing the most influence in a classification and label that as the bounding box
* Two aspects in particular were responsible for most of the quality improvement: primarily, active learning, and secondarily, per-channel energy normalization applied to the spectrograms (PCEN; Wang et al., 2017).
* The spectrogram classification component of the model was the architecture that Hershey et al. (2017) found best at detecting hundreds of audio event classes in YouTube videos. Specifically, we used a ResNet-50 convolutional neural network (He et al., 2016).
* Our annotated data clearly has both species and type of vocalization. What exactly do we want our classifier to focus on?
* PCEN and MEL normalization seem to be useful to old groups.
* The bounding boxes are made using 1200 FFT window length, which is the same size window length the 22 team used
* The VAE is not recommended.
* Maybe we just need to throw more data when training the CNN to make it generalizable and more accurate. We could also use a resnet architecture
* We could also just focus on background noise removal.
* This more closely aligns with what Object Detection is all about.
  + Its goal is to “find the objects in the image.” Not to “determine if the image contains objects.” (!!!! I DON’T LIKE THIS)
* Has the 24 team implemented all of the recommendations from the 22 team
* In particular, the Capstone Team suggested training an image recognition model (or using an existing one from a third-party) to screen spectrograms for the presence of humpback vocalizations before using our model to make predictions on the “humpback-positive” spectrograms.
* Ask about different tools for the ML pipeline
* Verify that the preprocessing steps are worth doing and implemented correctly
* Focus on a two-layer approach, first doing a multi-class (could start with a binary class) classification (with an unknown and no call category), then try to build bounding boxes in snippets with known calls
* Use the sage maker built in tools as a benchmark for our own model
* Not all the hydrophones are the same but they have the same sample rate, Monterey has an older hydrophone
* Exploratory analysis of the soundscape of the CA coast
* Scrips?? (Triton in MATLAB, has basic detection items and useful) Trying to implement RCNN for the package (in python)
* Large Hz range makes creating an overall useful model (computer vision) very difficult
* Constant Q transform over FT? Constant Q could make a larger Hz range spectrogram possible
* Could move away from computer vision into audio data transformers. Could leverage long term context
* Michaela thinks that the image normalization does not help! Test raw vs normalized
* Pretrained models are not trained on spectrograms, do they convert well?
* Enough data?
* We have to make decisions about what exactly the tool will look like
* Unsupervised to make a more general model
* Seems like generally a multiple-layer approach is a good idea
* Open soundscape in python
* Evaluating machine learning labels

Overall Goal: Build a data pipeline that processes full audio (.wav) files to show where events of interest are to focus research effort on useful time periods in the data (and ignore uninformative data).

Action plan:

1. Ensure the 2024 preprocessing pipeline works correctly and accurately
   1. Test on raw and processed data
2. Test the internal sage maker tools for their efficacy
3. Create a multi-class classifier for small snippets (5-10 seconds) of time
4. Create a bounding box model to work on the data identified by the classifier as part of a multilayered system
5. (OPTIONAL) Examine audio data transformers to glean additional context from the model

Note: This explicitly does not focus on understanding the types or contexts of the calls, leaving that work to the researchers. It will likely not be very explainable at first.

Previous teams have done a lot of work on the pipeline but failed to get a model that had worthwhile results. Focus on the results of model so that the output is as useful for human researchers as possible. Always keep track of what we are doing, journal about why we made the decisions we made.

!!! Sagemaker canvas is billed for having resources allocated to it, which is whenever you are logged in. Exiting the app is not enough, you must log out, which will kill any running model jobs you have, as well as any API endpoints you have setup.