

GDMS – Engine Detection

Car Engine Audio Classifier

Teams Members: Rayden Dodd, Ryan Hua, Valeria Pozo, Dylan Green, Austin Hall

Mentor: Dr. Daniel Connors | **Sponsors:** Scott Patterson, Ethan Brooks

SME: Dr. Sook Shin Ha

April 17, 2024

Problem Statement

- Show the possibility of identifying an engine's make from the sounds it produces while idling.
- Demonstrate engine audio has identifiable features.
- Will be used in the future to detect submarines underwater
- Intended as a proof of concept



Design Requirements

- Program is standalone and offline
- Constantly listening for idling engine
- System activates upon hearing engine sound
- Displays closest 3 car brands in GUI
- Uses narrow-band analysis to analyze audio
- Results generated within 5 minutes.



Data Acquisition

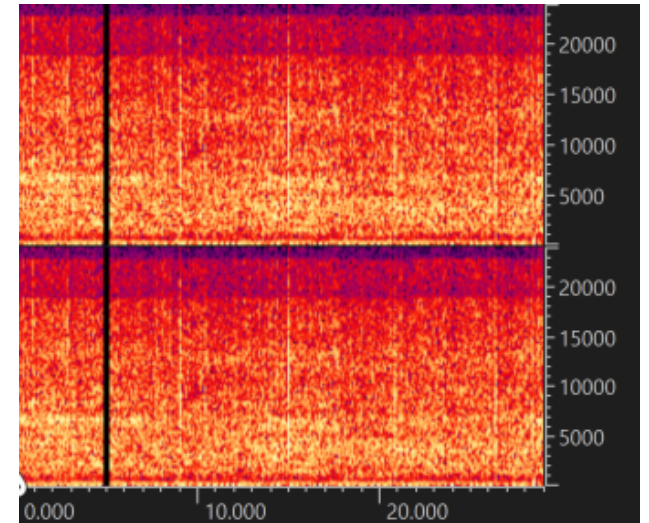
- Data collected both from real-world and online audio
 - Over 200 audio samples collected
- Audio samples saved in WAV files
 - Uncompressed
 - 48 kHz
 - 32-bit depth
- Sample length ranges from
 - 10-30 seconds long for engines from online data
 - 1 minute to 5 minutes for engines we recorded ourselves



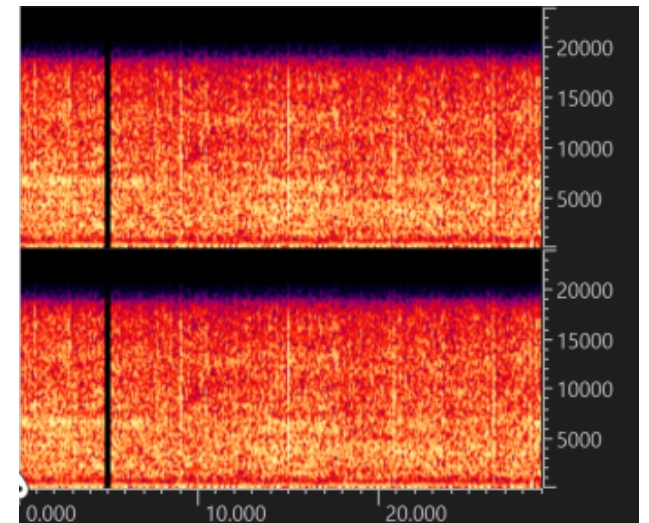
Our audio recording setup

Preprocessing

- Data augmentation
 - Random Noise
 - Gain Level Adjustments
 - Pitch Shifting
 - High Pass filtering
 - Low Pass filtering
- Applied each type of augmentation to each file
 - Resulted in over 1,000 engine audio samples
- Cleaned data by clipping out background noise, talking, etc.



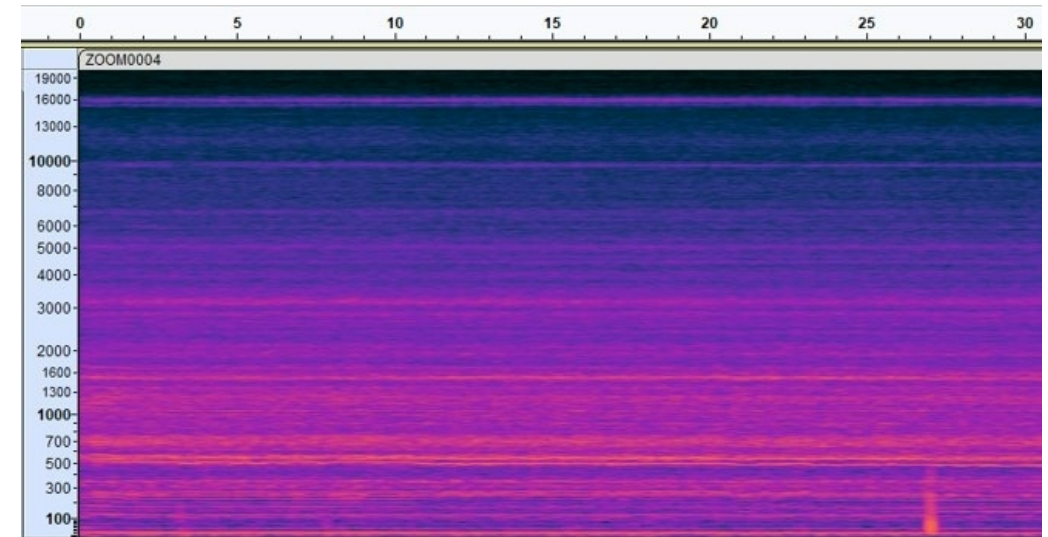
Spectrogram of Honda prelude
before augmentation above



Spectrogram of Honda prelude
after augmentation above

Feature Extraction

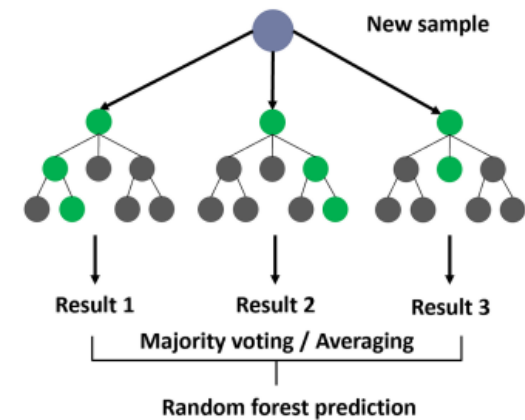
- Data segmentation
 - All audio split into 0.1-second segments
- MFCC extraction
 - Mel Frequency Cepstral Coefficient
 - Condensed representation of spectrogram
 - Closely related to how the human ear detects frequencies
 - Well suited for machine learning tasks



Spectrogram of 2008 Scion tC
audio

Engine Detector

- Used to detect when an engine has turned on
- Constantly listening
- Uses binary classifier model (Random Forest Classifier)
 - Alternates tried
 - Change in dB level detected (Too generic)
 - Rhythm detected (Not always rhythmic noises) (RPM is below human hearing)



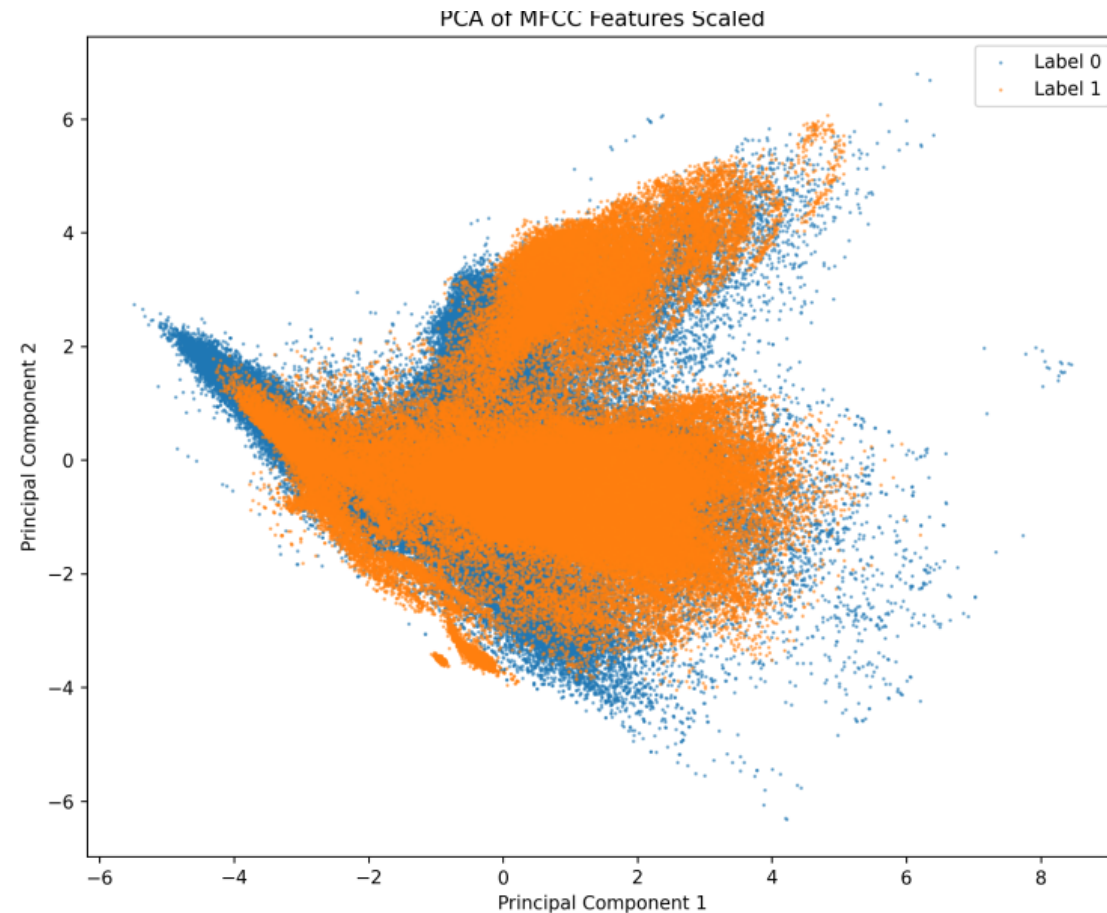
Engine Detection Training Set Creation

- Created a database to train the model on
 - Background (Not an engine, label 0)
 - This category involves anything that isn't an engine: wind, talking, walking, etc.
 - Engine (label 1)
 - Used our engine recordings as stated before
 - Balanced dataset - both classes have 148,050 1-second segments

Engine Detection Visualization

Principal Component Analysis
Reducing the 13 MFCCs to a 2D space

- Variance
 - Overall Variance 18,667.057
 - Label 1: 13,375.484
 - Label 0: 23,940.162



Engine Detection Training

- Tested 16 different binary classification models.
 - Used “Kfolds” cross-validation with 10 folds
 - “ExtraTrees” had the highest accuracy with 98.57%
 - Perceptron had the lowest accuracy with 71%
 - Optimized models to get the best hyperparameters

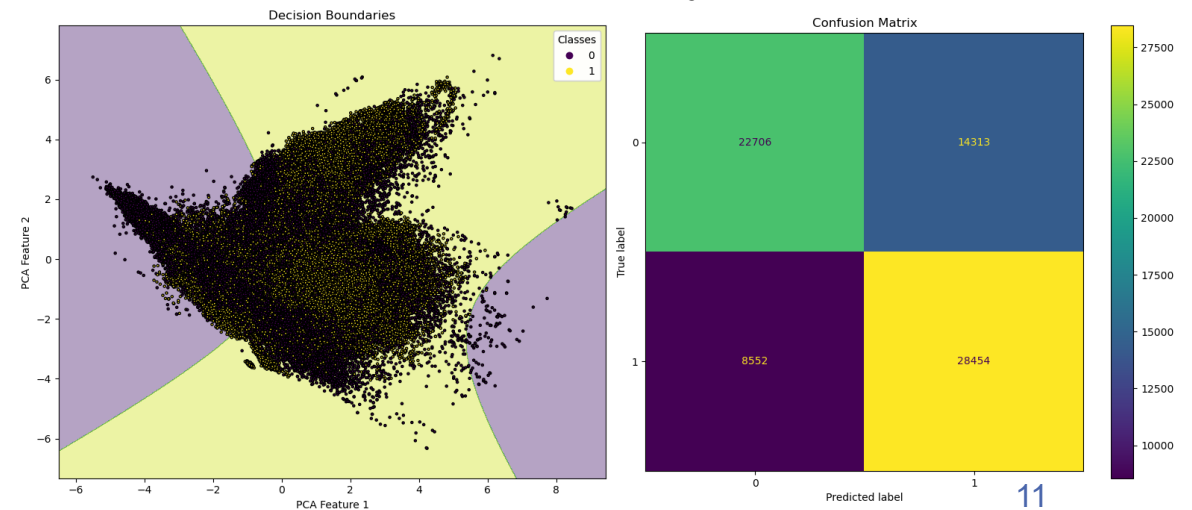
Binary Classifier Models used

GaussianNB
MultinomialNB
RandomForest
LogisticRegression
SVM
GradientBoosting
Ridge
PassiveAggressive
Perceptron
SGD
KNeighbors
DecisionTree
ExtraTrees
Bagging
AdaBoost
QuadraticDiscriminantAnalysis

Engine Detection Testing

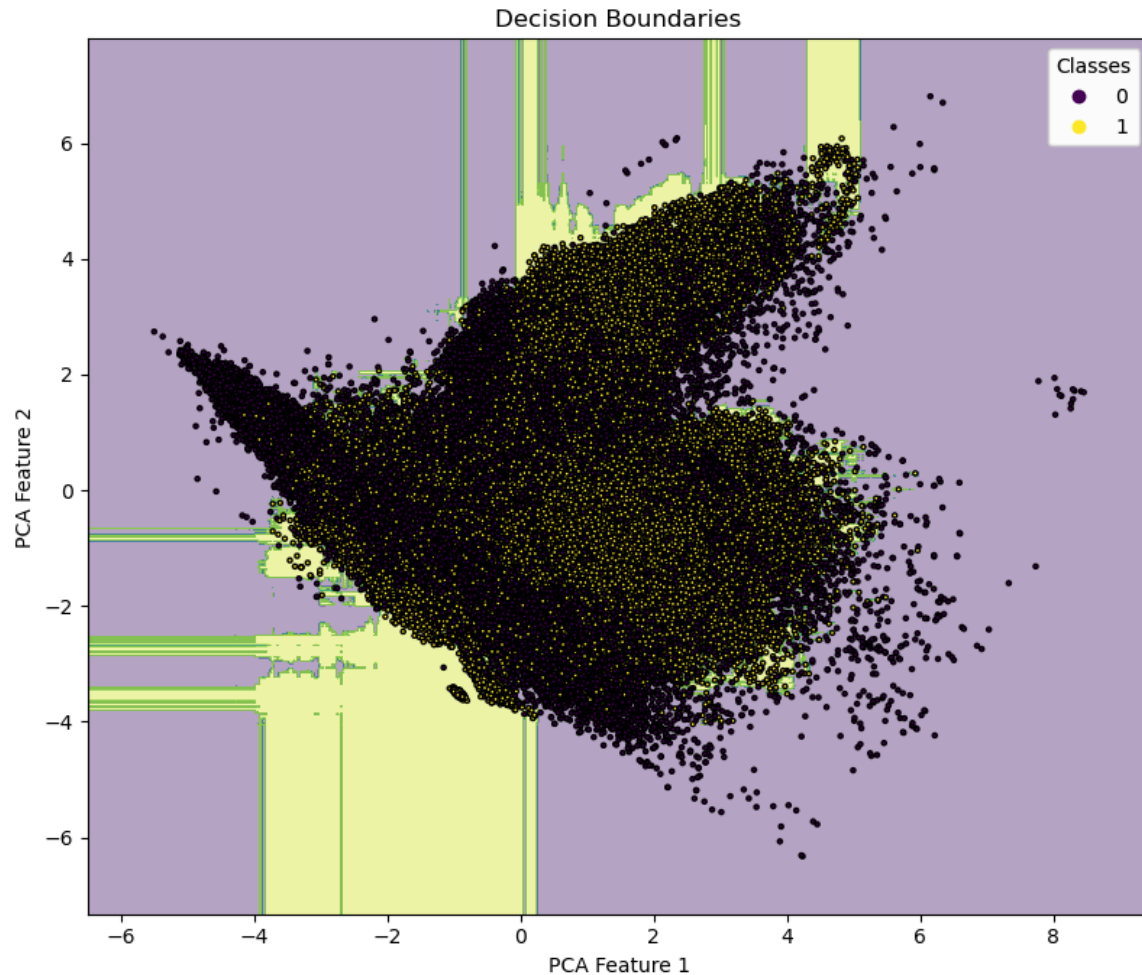
- Separately recorded validation set
 - Microphone at varying distances
 - 10 mins of car idling
 - 10 mins of ambient sounds
- Tested all models against the validation set
 - Quadratic Discriminant Analysis performed the best with an accuracy of 85.8
 - Model learned loud = engine
 - Got 10% accuracy on just loud music alone

Quad's Decision Boundary/ Confusion Matrix

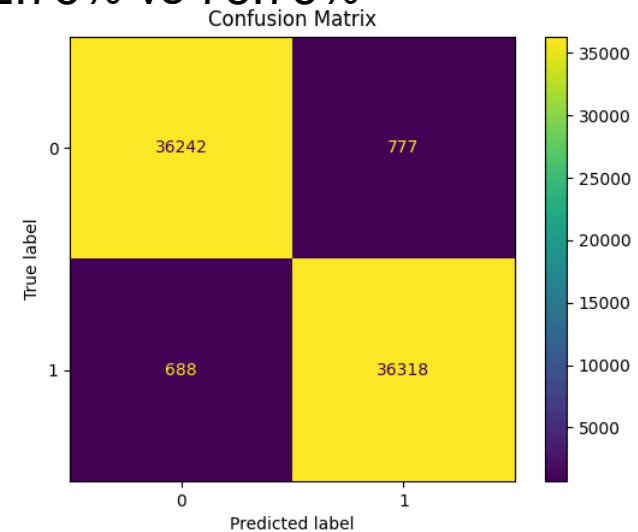


Not very complex boundary how is it getting so high?

Engine Detection Results

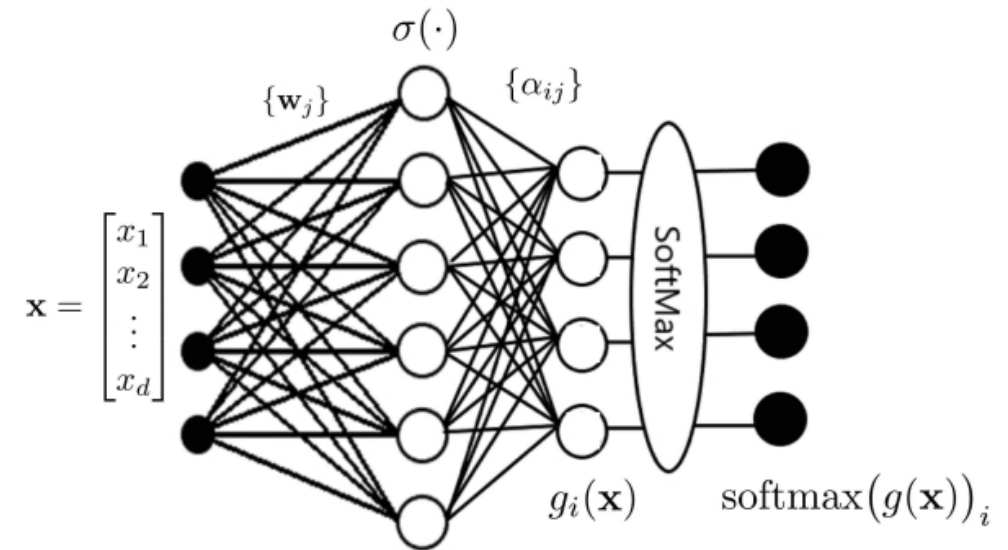


- We decided to go with Random Forest
 - Had the second-best accuracy in the validation set 81.7%
 - I picked it over AdaBoost as Random had a higher accuracy on engines 82.75% vs 78.75%



Engine Classifier Approach

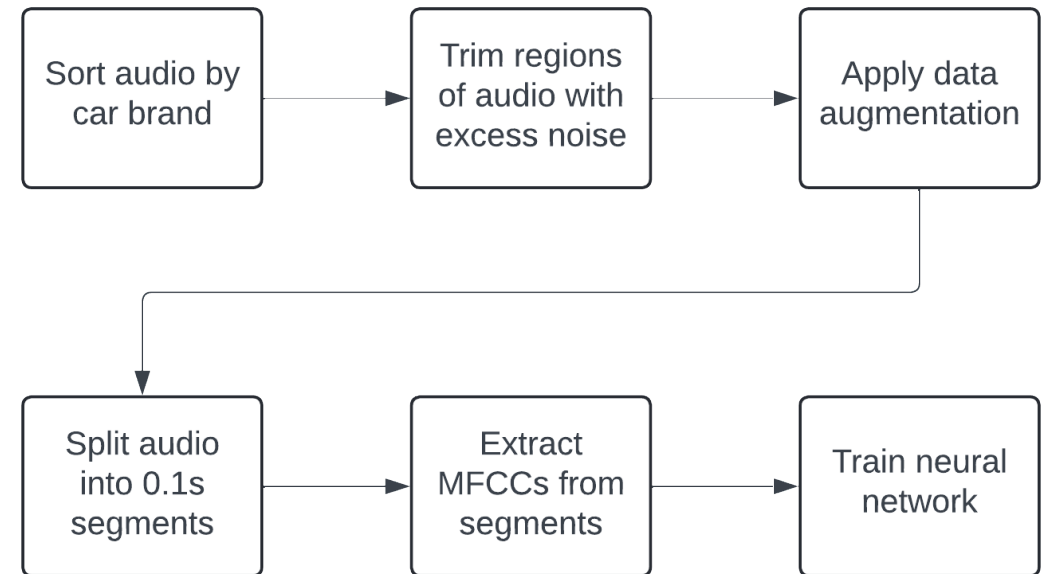
- Deep Neural Network
 - Classifies audio into top 10 car brands in the USA
 - MFCC vectors used as input
 - Outputs probability for each car brand
- Architecture
 - 3 hidden layers
 - 10 outputs
 - Dropout used to reduce overfitting



Neural network architecture example

Engine Classifier Training

- Steps to reduce overfitting
 - 0.1s audio segments reduce input dimensionality
 - Training stops after 5 epochs of no improvement
- Audio data split into training/testing sets
 - 80/20 split
 - Lets us see performance during training



Classifier training approach

Engine Classifier Results

- Exceeds target accuracy
 - 96% classification accuracy
- Data augmentation improved performance
 - Only 75% accuracy training on unaugmented audio
- Results produced within the required time
 - 8.5 seconds on average

		Predicted Brand									
		BMW	Ford	GMC	Honda	Hyundai	Jeep	Kia	Nissan	Subaru	Toyota
Actual Brand	BMW	91	0	0	0	0	1	1	0	0	2
	Ford	0	69	0	0	0	0	0	0	0	1
	GMC	0	0	102	1	0	0	0	0	2	5
	Honda	0	0	1	122	1	0	1	0	0	0
	Hyundai	1	0	0	1	84	0	4	0	0	0
	Jeep	0	0	0	1	0	84	1	1	0	3
	Kia	1	0	0	0	0	0	89	0	0	0
	Nissan	1	0	0	1	0	0	1	107	0	0
	Subaru	1	0	1	0	0	1	0	0	107	0
	Toyota	0	0	0	5	2	0	1	0	0	222

Engine classifier confusion matrix

Graphical User Interface

Most Likely Brands

Jeep

Brand 1: Jeep

Input Selection

Microphone Audio File

Select Microphone:


Blue Snowball Mono

Status: Ready

Start Analysis Stop Analysis

GMC

Brand 2: GMC



HONDA

Brand 3: Honda

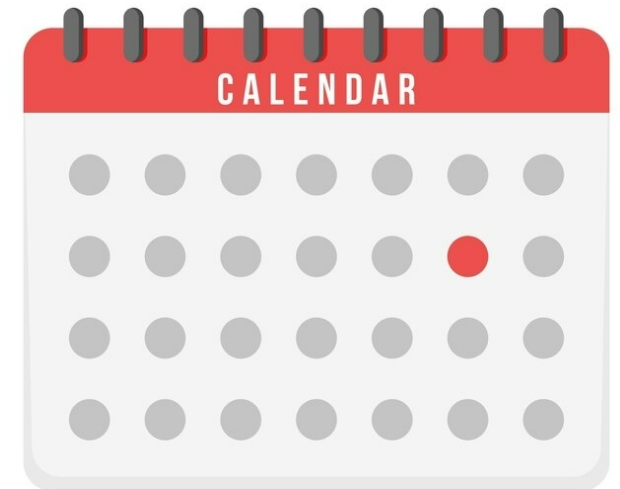
Challenges

- Collecting Data
 - Difficult to find datasets of engine idling
 - Had difficulty acquiring a balanced dataset
- Performance of the Machine Learning Model
 - Model prone to overfitting
 - Longer audio segments leads to high input dimensionality
 - Distance between engine and microphone affects model accuracy
- Preparing the Model
 - Tuning model parameters is time-consuming
 - Had difficulty working with a large audio dataset



Scheduled Milestones

- Initial Design Concept – Sept. 15, 2023
- Preliminary Design Review – Oct. 26, 2023
- Verification Test Plan – Nov. 12, 2023
- Demo of Engine Classifier with 4 brands – Nov. 28, 2023
- Critical Design Review – Dec. 1, 2023
- Detailed Design Documentation – Dec. 2, 2023
- Prototype Demo – Mar. 13, 2024
- Finished Engine Classifier - Apr. 5, 2024



Results

- System can run completely offline
- Neural network classifies with an accuracy of 96%
- System activates when an engine is detected
- Average time per engine identification: 8.5 seconds
- All project requirements met



Moving Forward

- Dataset
 - Balance total audio length between car brands
 - Include a non-car engine class (motorcycles, planes, etc.)
- Machine Learning Model
 - Add the ability to determine the car model along with the bra
 - Test other neural network architectures
 - Test effects of microphone distance on classification accurac



Acknowledgments

We want to thank the following people for supporting us throughout the project:

- Daniel Connors
- Sook Ha
- Scott Patterson
- Ethan Brooks

