Engine Detection Classification

Rayden R. Dodd, *IEEE*

***Abstract*—This Document provides the development process of an engine detection model designed as a binary classifier to be able to tell when the sound input is a car engine idling or something else. Through a rigorous approach of iterative training and validation the model achieves the detection of car engines. This paper also touches on the challenges encountered, sound variability and model comparison.**

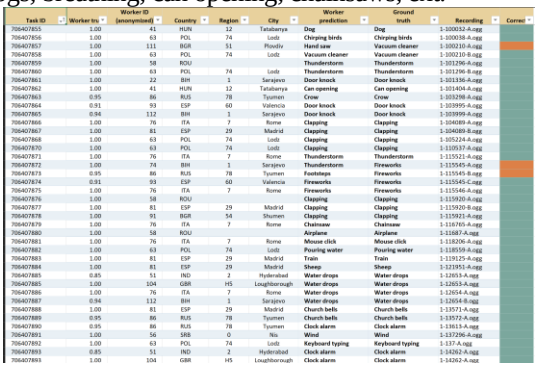
***Index Terms*—Engine Sound Classification, Binary Classification, MFCCs, Machine Learning, ExtraTrees.**

I. INTRODUCTION

For our Major Design Experience project we were tasked with creating an machine learning engine classifier model. The requirements for this model were that it had to constantly be listening to its surroundings and once it detected a car engine is idling it then has to run this sound clip through a machine learning model that is able to classify within 80% accuracy what brand of car it, as well as its next 2 brand predictions. This document will go over the engine detection part of this project. This includes talking about how the dataset for training was made, choice in model, optimization of the best model, and model testing.

II. DATA SET CREATION

To create the dataset first we needed sound recordings of stuff that wasn't car engine idling. So first I found a GitHub repo karolpiczak/ESC-50: ESC-50: Dataset for Environmental Sound Classification (github.com) that had a dataset of different environmental sounds. This dataset contained 2000 5 second audio clips of various noises including but not limited to frogs, breathing, can opening, chainsaws, ext.



Task ID	Worker ins	Worker ID (anonymized)	Country	Region	City	Worker prediction	Ground truth	Recording	Correct
706407855	1.00	41	HUN	12	Tatabanya	Dog	Dog	1-100032-A.ogg	
706407856	1.00	63	POL	74	Lodz	Chirping birds	Chirping birds	1-100038-A.ogg	
706407857	1.00	111	BGR	51	Plovdiv	Hand saw	Vacuum cleaner	1-100210-A.ogg	
706407858	1.00	63	POL	74	Lodz	Vacuum cleaner	Vacuum cleaner	1-100210-B.ogg	
706407859	1.00	58	ROU			Thunderstorm	Thunderstorm	1-101296-A.ogg	
706407860	1.00	63	POL	74	Lodz	Thunderstorm	Thunderstorm	1-101296-B.ogg	
706407861	1.00	22	BIH	1	Sarajevo	Door knock	Door knock	1-101336-A.ogg	
706407862	1.00	41	HUN	12	Tatabanya	Can opening	Can opening	1-101404-A.ogg	
706407863	0.95	86	RUS	78	Tyumen	Crow	Crow	1-103298-A.ogg	
706407864	0.91	93	ESP	60	Valencia	Door knock	Door knock	1-103995-A.ogg	
706407865	0.94	112	BIH	1	Sarajevo	Door knock	Door knock	1-103999-A.ogg	
706407866	1.00	76	ITA	7	Rome	Clapping	Clapping	1-104089-A.ogg	
706407867	1.00	81	ESP	29	Madrid	Clapping	Clapping	1-104089-B.ogg	
706407868	1.00	63	POL	74	Lodz	Clapping	Clapping	1-105224-A.ogg	
706407870	1.00	63	POL	74	Lodz	Clapping	Clapping	1-110137-A.ogg	
706407871	1.00	76	ITA	7	Rome	Thunderstorm	Thunderstorm	1-115521-A.ogg	
706407872	1.00	74	BIH	1	Sarajevo	Thunderstorm	Fireworks	1-115545-A.ogg	
706407873	0.95	86	RUS	78	Tyumen	Footsteps	Fireworks	1-115545-B.ogg	
706407874	0.91	93	ESP	60	Valencia	Fireworks	Fireworks	1-115545-C.ogg	
706407875	1.00	76	ITA	7	Rome	Fireworks	Fireworks	1-115546-A.ogg	
706407876	1.00	58	ROU			Clapping	Clapping	1-115920-A.ogg	
706407877	1.00	81	ESP	29	Madrid	Clapping	Clapping	1-115920-B.ogg	
706407878	1.00	91	BGR	54	Shumen	Clapping	Clapping	1-115921-A.ogg	
706407879	1.00	76	ITA	7	Rome	Chainsaw	Chainsaw	1-116765-A.ogg	
706407880	1.00	58	ROU			Airplane	Airplane	1-116874-A.ogg	
706407881	1.00	76	ITA	7	Rome	Mouse click	Mouse click	1-118206-A.ogg	
706407882	1.00	63	POL	74	Lodz	Pouring water	Pouring water	1-118559-A.ogg	
706407883	1.00	81	ESP	29	Madrid	Train	Train	1-119125-A.ogg	
706407884	1.00	81	ESP	29	Madrid	Sheep	Sheep	1-121951-A.ogg	
706407885	0.85	51	IND	2	Hyderabad	Water drops	Water drops	1-12653-A.ogg	
706407885	1.00	104	GBR	H5	Loughborough	Water drops	Water drops	1-12653-A.ogg	
706407886	1.00	76	ITA	7	Rome	Water drops	Water drops	1-12654-A.ogg	
706407887	0.94	112	BIH	1	Sarajevo	Water drops	Water drops	1-12654-B.ogg	
706407888	1.00	81	ESP	29	Madrid	Church bells	Church bells	1-13571-A.ogg	
706407889	0.95	86	RUS	78	Tyumen	Church bells	Church bells	1-13572-A.ogg	
706407890	0.95	86	RUS	78	Tyumen	Clock alarm	Clock alarm	1-13613-A.ogg	
706407891	1.00	56	SRB	0	Nis	Wind	Wind	1-137296-A.ogg	
706407892	1.00	63	POL	74	Lodz	Keyboard typing	Keyboard typing	1-137-A.ogg	
706407893	0.85	51	IND	2	Hyderabad	Clock alarm	Clock alarm	1-14262-A.ogg	
706407893	1.00	104	GBR	H5	Loughborough	Clock alarm	Clock alarm	1-14262-A.ogg	

Fig. 1. This is a sample of the raw dataset with different classes shown with the audio file and where the audio was taken from.

Next, I needed to add our car engine recordings into the dataset. First of all, our recordings needed cleaning. For most of our car engine recordings the first 1-2 mins is just background noise until we turn the car on. Thus, using a python script that could detect the sudden change from a low volume recording of nothing to a loud engine starting I was able to cut the audio files into two background noise and engine idling. Because of this we had more environmental noise to add to our data set as well. Now that the files were cut to only include engine idling, we had to cut down each video into 5 second segments as the rest of the data set were also 5 second audio clips. Using another python code, I was able to chop all of the audio clips into 5 second segments leaving us with 2000 unique car engine idling clips. This was perfect as we had 2000 non engine idling clips and thus this made out database balanced which is beneficial in model training.

III. FEATURE EXTRACTION

To be able to train a model on this database, we first needed to convert the .WAV and .ogg audio files into numerical features that a machine learning algorithm can interpret. To this end, we utilized Mel-Frequency Cepstral Coefficients (MFCCs), which are a type of feature widely used in the processing of audio signals, particularly in the context of speech and music. MFCCs effectively represent the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. While spectrograms also provide a visual and numerical representation of the frequency spectrum over time, MFCCs are used to distill this information into a smaller set of features that capture the most important aspects of the sound for the purpose of auditory perception. This condensation into a more compact representation makes MFCCs more amenable to machine learning models, which can efficiently process and classify audio data based on these reduced features without a significant loss of information critical to distinguishing between different sounds or voices.

By employing MFCCs, we transformed the audio features into structured NumPy arrays, enabling their integration with scikit-learn machine learning models.

IV. SELECTION OF BINARY CLASSIFCATION MODEL

A. Testing Methodology

When coming up with a binaray classification model to train our data on our Subject Matter Expert (SME) suggested that we use Random Forest. To test this I used a list of different binary classification models to compare and see which model would give the highest accuracy. These models include:

1) Adaptive Boosting Classifier (AdaBoostClassifier)
2) Bagging Classifier (BaggingClassifier)
3) Decision Tree Classifier (DecisionTreeClassifier)

4) Extremely Randomized Trees Classifier (ExtraTreesClassifier)
5) Gaussian Naive Bayes (GaussianNB)
6) Gradient Boosting Classifier (GradientBoostingClassifier)
7) K-Nearest Neighbors Classifier (KNeighborsClassifier)
8) Logistic Regression (LogisticRegression)
9) Multinomial Naive Bayes (MultinomialNB)
10) Passive Aggressive Classifier (PassiveAggressiveClassifier)
11) Perceptron (Perceptron)
12) Quadratic Discriminant Analysis (QuadraticDiscriminantAnalysis)
13) Random Forest Classifier (RandomForestClassifier)
14) Ridge Classifier (RidgeClassifier)
15) Stochastic Gradient Descent Classifier (SGDClassifier)
16) Support Vector Machine (SVM)

To compare these models fairly, I employed two distinct cross-validation techniques: Leave One Out Cross Validation (LOOCV), which splits the dataset such that each instance serves once as the test set while the remainder forms the training set, ensuring every data point is utilized for validation, and K-Folds Cross Validation, dividing the dataset into for our case 10(K) equal parts, or "folds," and then iteratively uses one fold for testing and the remaining for training, with results averaged for thorough evaluation. After each iteration of LOOCV and K-Folds, I recorded the Accuracy, Recall, Precision, and F1 metrics, allowing these performance indicators to be averaged at the end of training to provide a comprehensive assessment of each model's effectiveness.

B. Results

The results of this showed that the best model out of the 16 different models was ExtraTreesClassifer which had a 99.02% accuracy in the LOOCV and in second place Random Forest with 98.25% accuracy. The worst performing model was MultinomialNB 53.03%. The ExtraTreesClassifier performed better, most likely due to the fact that it makes use of extreme randomness in selecting splits for each node in the decision trees. Unlike Random Forest, which seeks the most optimal split among a random subset of the features, ExtraTreesClassifier randomly selects the split points, resulting in a higher level of model variance and diversity. This approach can be particularly effective in complex classification tasks like distinguishing engine sounds, as it may better capture the nuanced differences between classes. On the other hand, the MultinomialNB performed so poorly because it operates under the assumption that all features are independent of each other—an assumption that doesn't hold true for audio data. Audio signals, especially features extracted for machine learning like MFCCs, exhibit significant correlation between features, which likely led to the algorithm's inability to accurately model the data and resulted in its low accuracy.

Comparing these two models against each other on the same set of data we get the following results:

TABLE I
F1 METRICS FOR EXTRATREES

Metric	Class 0	Class 1	Macro Avg	Weighted Avg
Precision	0.98	0.98	0.98	0.98
Recall	0.99	0.98	0.98	0.98
F1-Score	0.98	0.98	0.98	0.98
Support	550	448	998	998

TABLE II
ACCURACY PERCENTAGES FOR EXTRATREES

Overall Accuracy	98.30%
K-Fold Accuracy	0.99 (+/- 0.01)
LOOCV Accuracy	0.99 (+/- 0.20)

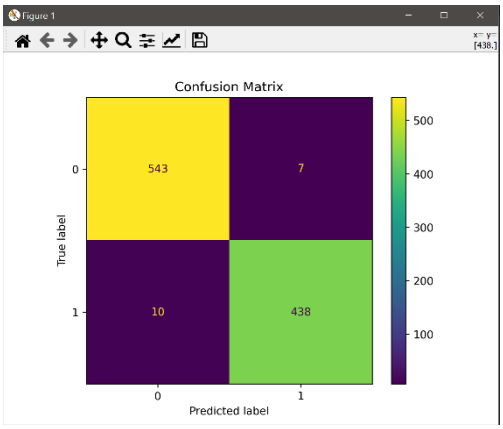


Fig. 2. Confusion matrix for ExtraTrees

TABLE III
F1 METRICS FOR RANDOM FOREST

Metric	Class 0	Class 1	Macro Avg	Weighted Avg
Precision	0.98	0.97	0.97	0.97
Recall	0.98	0.97	0.97	0.97
F1-Score	0.98	0.97	0.97	0.97
Support	550	448	998	998

TABLE IV
ACCURACY PERCENTAGES FOR RANDOM FOREST

Overall Accuracy	97.49%
K-Fold Accuracy	0.98 (+/- 0.01)
LOOCV Accuracy	0.98 (+/- 0.27)

Fig. 3. Confusion matrix for Random Trees

These results show that overall ExtraTrees is slightly better than Random trees due to the better accuracy and F1-scores.

V. Model Optimization

To optimize the ExtraTree model I used the sklearn documentation page to find all of the available parameters for the model and using GridSearch Cross Validation built into sklearn after some time it showed the best parameters which were : n_estimators=250, max_features = 7, min_samples_leaf =1, random_state=42, n_jobs=-1. This brought the single run accuracy from 98.3% to 98.4% so only a small bump in performance

V. Conclusion

In conclusion, developing and comparing binary classification models to detect engine sounds has shown us how challenging and intricate this task can be. The ExtraTreesClassifier stood out with its impressive accuracy and ability to avoid overfitting, demonstrating the value of using a more random approach in decision trees. This project didn't just test different machine learning models; it also gave us insights into how to fine-tune these models for better results. Additionally, the poor performance of the Multinomial Naive Bayes model highlighted the need to choose the right machine learning strategy, considering the data's specific traits, like the connection between audio features. Our careful work on preparing the data, picking out features, testing models, and refining them has set a solid foundation for future work in audio classification. This could open up new possibilities in automated engine diagnostics and more. This project was an important part of our major design experience, contributing useful knowledge and techniques to the field of machine learning.