

Introduction

This is a document for Audio Processing project work. Work topic is default (option 1) and was given in the project work instructions. The idea is described below and I'm working alone.

In this work I have collected audio samples from cars and buses. The goal is to develop a machine learning model capable of predicting these vehicles from sound. Also do feature extraction for audio signals for deeper look into audio files and find the correct techniques for model. Project is done in five different stages which are the headers in this document.

Data description

There were many different vehicles to choose from in the project work instructions. I chose two classes which are a car and a bus. The main reason for using these classes is that for me it was easiest to get all recordings. All collected audio samples are uploaded to [Freesound.org](https://freesound.org) and the link is provided to excel sheet. All my uploaded samples are free to use for every user with zero license.

Car audio samples are all recorded on the same day. I went for a walk late November in Haaga (Helsinki) and recorded passing cars. Most of the records have only one car passing by, and I tried to minimize background noises. I did listen and approved all the recordings before submitting them to Freesound.

Weather conditions for car samples were quite optional. It was early winter in Finland, so temperature was between minus five and minus eight degrees. There was slight wind but not so heavy that my recorded got it. There was one slight problem in a few samples where you could hear the studded tires, but I don't think it will mess up my modeling.

Bus audio samples are not recorded on the same day. I saw many students collecting idling bus sounds, but I wanted to use passing by samples. I have recorded buses at multiple locations in Tampere. Most of the samples are from Hervanta but a few of them are from Hatanpää. I tried as much as possible to get a sample where just a single bus passed by without any cars close to it. I succeeded in this at the most samples.

Weather conditions for bus samples can differ slightly from sample to another. The temperature is between minus five to minus fourteen. There was slight wind at every recording day, but it did not bother recordings. One noticeable thing was that as most of the buses are electrical vehicles, the sounds of them idling and going forward no-throttle are much less noisy.

I recorded 20 samples for each class. At my project I used also audio samples from other students who have given permissions to use them. I use my own samples as test data, randomly chosen students samples as validation data and I downloaded a total of 100 samples from multiple students for my training data.

Feature extraction

Feature extraction is an important part of the classification task. We need extraction to transform original data into a numerical format. This way machine learning models get better results. We can also make models better by choosing suitable features for our needs. At first, I chose to extract MFCCs, Mel spectrogram, Zero Crossing Rate and RMSE. Later, I added Spectral Centroids and Chromagram. Most of the features are calculated with Librosa libraries.

Mel-Frequency Cepstral Coefficients (MFCCs) is a method that transforms the power spectrum to a small number of coefficients representing power. It gives a better representation of sound as it approximates the human auditory system more closely. MFCCs are a very used feature in audio processing. For example, we can see that Bus audios have more positive MFCCs in lower frequencies.

Mel spectrogram is derived from the MFCCs and gives us a power spectrogram where frequencies are mapped to Mel scale. With Mel Spectrogram we can understand Mel Spectrogram can be used as feature along with MFCCs. Analyzing Mel spectrograms, we can see that results are quite similar as in MFCCs but visually clearer; Bus audios have lower Mels than car. You can see it by inspecting the warm colors in the plot.

Zero crossing rate measures the rate at which the audio signal changes from positive to negative. For Car and Bus signals when a car is passing by the plot is quite similar, there are few differences due to engine sounds. This feature gives the model more accurate numbers when a vehicle is passing-by versus idling.

Root Mean Square Energy (RMSE) measures the overall energy amplitude over time of the audio signal. Roughly corresponds to how loud the signal is. I have used frame length of 512 and hop length of 256. I have tested values and was happy with the results. This feature is especially good for recognizing passing-by vehicles.

Spectral centroids are extracted from the spectrogram. They represent the spectrum's center of mass, indicating where the energy is concentrated.

Chromagram is representation of the pitch classes in different frequencies over time. The pitch distribution of the Bus samples is very stable and almost a clear line in F and E pitches. Car samples are more intense, and sharper compared to Bus.

Model selection

Data has been split into training, validation and test data. I used my own recordings as test data, randomly chosen students recordings as validation and test data contains randomly selected 100 samples per class. Combined my training data contains 200 samples, validation data 40

samples and test data 40 samples. I chose my own recordings as testing data due to the reason that I know for sure what kind of samples my models can predict.

For models, I used simple binary classifiers such as Logistic Regression, K-Nearest Neighbours, Support Vector Machine and Random Forest.

I chose these models since my dataset is quite small and my opinion is that I don't need to fit these more time-complex models such as DNN. For instance, SVC and kNN are particularly well suited for smaller datasets.

After training models without parameters and inspecting the results, I used GridSearchCV library from Sklearn to enhance results with aid of hyperparameters. I set each model their own grid paramers and tried to find the best parameters for my needs. I aimed to get better results for accuracy.

Results

Calculated results for models are shown in the picture below. They are sorted by Accuracy in decreasing order. The best model is Random Forest Classifier with 82.5 % accuracy. The grid search decreased accuracy with every model. For kNN the difference between Accuracy and Grid Accuracy was the smallest. To summarize the results, I would say Random Forest and SVC came out with pleasant results. Logistic Regression classifiers results were decent and kNN did not work at all for this task. Grid search did not work at all, and I do not think grid results should be taken to count.

| | Model | Accuracy | Grid Accuracy |
|---|---------------------|----------|---------------|
| 3 | Random Forest | 82.5 | 50.0 |
| 0 | SVC | 75.0 | 35.0 |
| 2 | Logistic Regression | 57.5 | 42.5 |
| 1 | kNN | 47.5 | 45.0 |

Image 1: Results for models

Conclusions

Models can predict cars and buses with great accuracy. I think the 82.5 % accuracy is a good result for this size of dataset. It seems that features are successfully extracted and selected features are good picks for my project.

Some of the results were good and some of them were unexpected. When trying to get GridSearch to work with validation data I faced troubles. When fitting the validation data for grid the results were terrible. When I looked up the validation data, I noticed that it differs a lot from test data (my own recordings). Most of the car samples were recorded from far away and some recordings were only from an idling car. My models couldn't predict cars at all. In pictures 2 and

3 are shown more specific results on two best classifiers. 0 for cars and 1 for buses. There you can see that the f1-scores for cars were close to zero.

| | | | | |
|-------------------|-----------|--------|----------|---------|
| SVC results: | | | | |
| | precision | recall | f1-score | support |
| 0 | 0.73 | 0.80 | 0.76 | 20 |
| 1 | 0.78 | 0.70 | 0.74 | 20 |
| accuracy | | | 0.75 | 40 |
| macro avg | 0.75 | 0.75 | 0.75 | 40 |
| weighted avg | 0.75 | 0.75 | 0.75 | 40 |
| Accuracy: 75.00 % | | | | |
| ----- | | | | |
| grid results: | | | | |
| | precision | recall | f1-score | support |
| 0 | 0.12 | 0.05 | 0.07 | 20 |
| 1 | 0.41 | 0.65 | 0.50 | 20 |
| accuracy | | | 0.35 | 40 |
| macro avg | 0.27 | 0.35 | 0.29 | 40 |
| weighted avg | 0.27 | 0.35 | 0.29 | 40 |
| Accuracy: 35.00 % | | | | |

Image 2: SVC results

| | | | | |
|-----------------------------------------|-----------|--------|----------|---------|
| Random Forest results: | | | | |
| | precision | recall | f1-score | support |
| 0 | 0.78 | 0.90 | 0.84 | 20 |
| 1 | 0.88 | 0.75 | 0.81 | 20 |
| accuracy | | | 0.82 | 40 |
| macro avg | 0.83 | 0.82 | 0.82 | 40 |
| weighted avg | 0.83 | 0.82 | 0.82 | 40 |
| Accuracy: 82.50 % | | | | |
| ----- | | | | |
| Random Forest with Grid Search results: | | | | |
| | precision | recall | f1-score | support |
| 0 | 1.00 | 0.00 | 0.00 | 20 |
| 1 | 0.50 | 1.00 | 0.67 | 20 |
| accuracy | | | 0.50 | 40 |
| macro avg | 0.75 | 0.50 | 0.33 | 40 |
| weighted avg | 0.75 | 0.50 | 0.33 | 40 |
| Accuracy: 50.00 % | | | | |

Image 3: Random Forest results

Results can be improved in many ways. Since the feature extraction works in the wanted way partly, I would focus the improvements on audio data quality. Training data now is 100 for each class, that could be a much bigger number. Same goes for validation and test data. When inspecting the training data, I realized that they all are not that good for my needs. Some samples may have little back noises, and some are recordings from idling vehicles.

My goal was to predict passing-by vehicles so the first step to improving I would go through used audio data. Secondly, I would take a closer look at feature extraction.

In image 4 are shown Receiver Operating Characteristic curves for SVC and Random Forest (left) and Confusion matrixes (right).

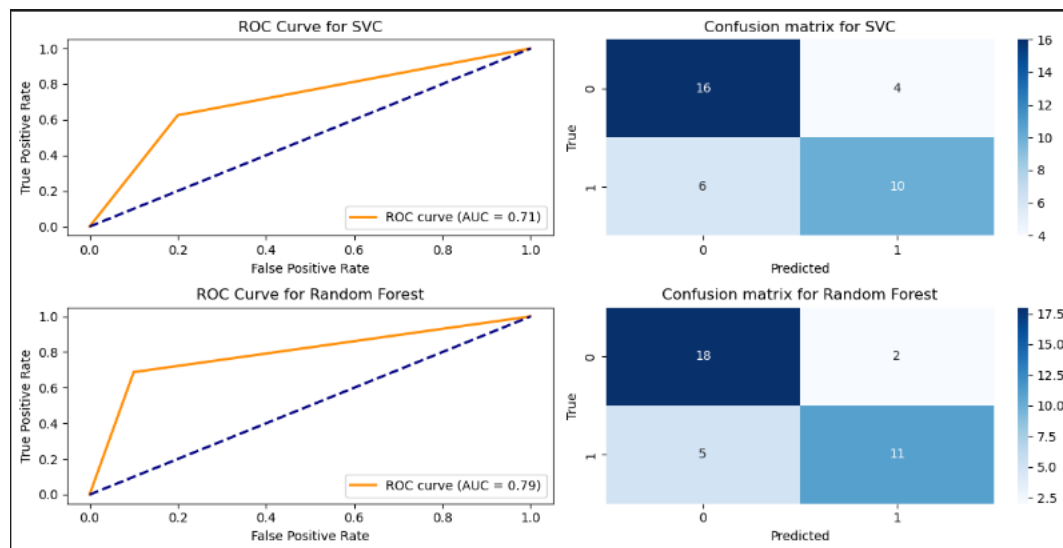


Image 4: Roc curves and confusion matrixes for SVC and Random Forest