Assignment 2 - ML

Rishi Pendyala - 2022403

SECTION A

Section A.

1 (a) let us consider

D : event that the company issues a dividend.

~D : the event that the company doesn't issue a divid...

P : profit increase percentage.

It's given that :

$P(D) = 0.8$ and $P(\sim D) = 0.2$

If dividends are issued

Profit increase.

$\hookrightarrow$ Avg increase = 10%

$\hookrightarrow$ Variance = 36%
(σ^2)

Likelihood

$$\hookrightarrow P(P = x \mid D) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2(\sigma^2)}}$$

$$P(P = 4 \mid D) = \frac{1}{\sqrt{2\pi(36)}} e^{-\frac{(4-10)^2}{2(36)}}$$

Likelihood of a 4% incr given the comp. issues a divided.

$$P(P = 4 \mid D) = \frac{1}{6\sqrt{2\pi}} e^{-1/2}$$

$$P(P = 4\% \mid \sim D) = \frac{1}{6\sqrt{2\pi}} e^{-\frac{(4-0)^2}{2(36)}}$$

If the dividends were not issued

$\hookrightarrow$ Avg profit increase = 0%

$$\Rightarrow P(P=4\% \mid \sim D) = \frac{1}{6\sqrt{2\pi}} e^{-2/4}$$

Applying Baye's Theorem:

$$P(D \mid P=4\%) = \frac{P(P=4\% \mid D) \cdot P(D)}{P(P=4\%)}$$

$$P(P=4\% \mid D) = \frac{1}{6\sqrt{2\pi}} e^{-1/2} = \frac{0.606}{6\sqrt{2\pi}} = \underline{0.0404}$$

$$P(P=4\% \mid \sim D) = \frac{1}{6\sqrt{2\pi}} e^{-2/4} = \frac{0.18}{15} = \underline{0.053}$$

$$P(P=4\%) = ((0.404) \times (0.8)) + ((0.2) \times (0.053))$$

$$P(P=4\%) = \underline{0.4292}$$

Thus,

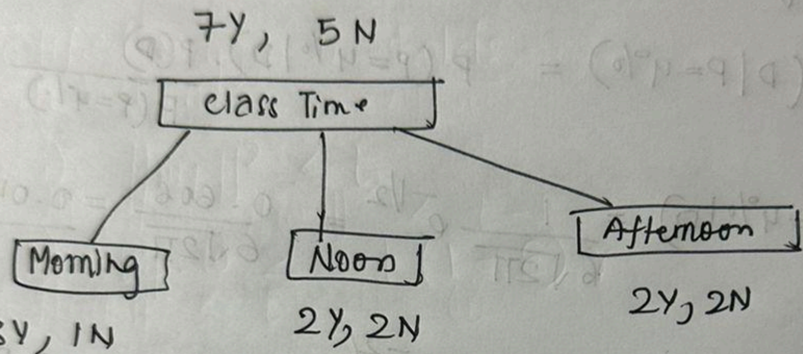
$$P(D \mid P=4\%) = \frac{(0.8) \times (0.0404)}{(0.04292)} = \underline{\underline{0.75}}$$

Thus, the company has a 75% prob that it will issue a dividend this year according to the trends given.

1(b). Building the decision tree for the table using IG

Using the ID3 info gain,
top-down.

Starting with class time:-



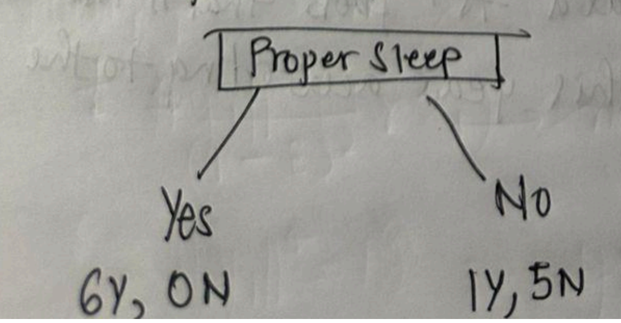
$$IG = H(Y) - H(Y|X)$$

$$\text{lowest } H(Y|X) = \sum_{r=1}^{n} \frac{|S_r|}{|S|} \times (\text{cross-entropy of each node})$$

For class time:

$$H(Y|X) = 0.93$$

For Proper Sleep.



$$H(Y|X) = \frac{1}{2} \times \left(-\log \frac{1}{2}\right)$$

$$+ \frac{1}{2}\left(-\frac{1}{6}\log \frac{1}{6} - \frac{5}{6}\log_2 \frac{5}{6}\right)$$

$$H(Y|X) = 0.32$$

Q:

For weather:

$$\text{Weather} \to \text{Cool},\ \text{Rain},\ \text{Hot}$$

Cool: 4Y, 1N

Rain: 0Y, 2N

Hot: 3Y, 2N.

$$\frac{5}{12}\left(-\frac{1}{5}\log\frac{1}{5} - \frac{4}{5}\log\frac{4}{5}\right) + \frac{5}{12}\left(-\frac{3}{5}\log\frac{3}{5} - \frac{2}{3}\log\frac{2}{5}\right)$$

$$+ \frac{2}{12}(0) = \underline{\underline{0.7}}.$$

Best Split: Mod. Proper sleep > lowest entropy

To find 2^{nd} best split we use the same method.

Proper Sleep

No | Yes

1Y, 5N | 6Y, 0N

Morning | Noon | Afternoon

1Y, 1N | 2N | 1N.

Again checking the $H(Y|X) = -\frac{1}{3}\left(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right)$

$$H(Y|X) = \frac{1}{3}$$

For weather:

Proper Sleep

No | Yes

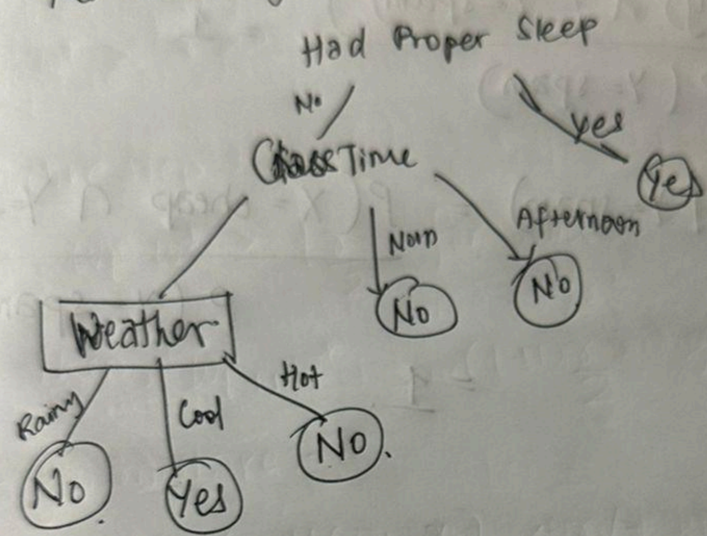
Weather

Raing | Cool | Hot

2N | 1Y 1N | 2N.

which will give us the exact same distribution as the last split.

∴ we take Class Time as the splitting attribute and the remaining attribute is weather. so the tree is



1(c). Given :-

(i) $S \rightarrow$ labelled dataset, that is linearly seperable by a margin γ.
(ii) All examples normalized to Euclidean length of 1.
(iii) Update on classification mistakes & margin mistake $\left(\text{margin} < \frac{\gamma}{2}\right) \rightarrow$ special perception algorithm.

To prove:
(i) No. of updates is almost $8/\gamma^2$
(ii) Derive mistake bound when margin threshold is $(1-\varepsilon)\gamma$.

(iii) $w_t \rightarrow$ wt of vector after t updates
$w^* \rightarrow$ optimal wt. vector
(given $\|w^*\| = 1$)

$X \rightarrow$ input vector ($\|X\| = 1$)
$y \rightarrow$ true label ($y \in \{-1, 1\}$).

Proof:-
On each update:

$$w_t + 1 = w_t + yx, \quad \|x\| = 1.$$

On squaring

$$\|w_{t+1}\|^2 = \|w_t + yx\|^2$$

$$\|w_{t+1}\|^2 = \|w_t\|^2 + \|yx\|^2 + 2yxw_t$$

Now, since we assumed points are misclassified.

Thus,
$$2y x w_t < 0$$

$$\|w_{t+1}\|^2 \leq \|w_t\|^2 + 1.$$

Now,

Induction!

$$\|w_t\|^2 - \|w_0\|^2 \leq t-1$$

$$\|w_t\|^2 \leq t \quad \text{——} \quad (1)$$

Now, to prove:
$$w_t \cdot w \geq t\frac{r}{2}$$

when $t=0$; $w_0 = y_1 x_1$

Also, $y_i (w^* x_i) \geq r$. $\longrightarrow$ Given by min. margin.

So $y(w^* x_1) \geq r$

Multiplying both sides with y_1,

$$\|y_1\|^2 (w^* x_i) \geq y_1 r$$

$$\left(\because \|y_1\|^2 = 1 \Rightarrow y_i \in \{-1,1\}\right)$$

$\because w^* x_1 \geq y_1 \gamma$

$w_0 w^* \geq y_i x_i w^*$

$w_0 w^* \geq y_i^2 \gamma$

$w_0 w^* \geq \gamma > 0 \Rightarrow w^* w_0 \geq \frac{\gamma}{2}.$

<u>Inductive step:</u>

$w_t \cdot w^* \geq t\frac{\gamma}{2}$. for some $t>0$.

then, $w_{t+1} w^* \geq (t+1)\frac{\gamma}{2} > \frac{t\gamma}{2}.$

Margin mistake:-

$0 < y(w_t \cdot x) < \gamma/2$ (given)

$w_{t+1} = w_t + yx.$

$(w_{t+1}) w^* \geq w_t w^* + \gamma \quad [\because y(w^* x) \geq \gamma]$

Also, $w_t w^* \geq t\frac{\gamma}{2}.$

$w_{t+1} w^* \geq t\frac{\gamma}{2} + \gamma$

$\geq t\frac{\gamma}{2} + \frac{\gamma}{2} + \frac{\gamma}{2}.$

$\geq t\frac{\gamma}{2} + \frac{\gamma}{2} \quad (\because \frac{\gamma}{2} > 0)$

each)

thus $\Rightarrow w_{t+1} \cdot w^* \geq (t+1)\frac{\gamma}{2}$

Hence proved.

Thus $\Rightarrow w_t \cdot w^* \geq t\frac{\gamma}{2}$ —— ②

Using Cauchy Swastz Inequality:

$\Rightarrow (a \cdot b)^2 \leq \|a\|^2 \, \|b\|^2$

$\Rightarrow (w_t \cdot w^*)^2 \leq \|w^t\|^2 \cdot \|w^*\|^2$

$\Rightarrow (w_t \cdot w^*)^2 \leq (t-1)$ $\quad [\because \|w_t\|^2 < t]$

Also, $w_t \cdot w^* \geq t\frac{\gamma}{2}$ from eqn ②

So, $\left(t\frac{\gamma}{2}\right)^2 \leq t$.

$t^2\frac{\gamma^2}{4} \leq t$. $\Rightarrow$ $\boxed{t \leq 4/\gamma^2}$ $\quad (\because \gamma^2 > 0)$

We have to account for
- classification mistakes
- margin mistakes

we'll multiply by 2 to account for both mistakes,

$$t \leq \frac{8}{\gamma^2}$$

Modified margin thresholds:-

i) $\|w_t\|^2 \leq t$

ii) $w_t \cdot w \geq t(1-\varepsilon)\gamma$

Proof for (i)

Each update incr. $w w^*$ by at least $(1-\varepsilon)\gamma$.
Thus after t updates:

$$w_t \cdot w^* \geq t(1-\epsilon)\gamma$$

Applying Cauchy Swartz:

$$[t(1-\varepsilon)\gamma]^2 \leq t$$

$t \leq \frac{1}{(1-\varepsilon)^2}\gamma^2 \longrightarrow$ new mistake bound.

1 (c

(i)

(ii)

(iii

1(d) For Spam:

(a) $P(X = buy \mid Y = spam)$

$$\frac{P((X = buy) \cap (Y = spam))}{P(Y = spam)} = 1$$

$$P(X = cheap \mid Y = spam) = \frac{P(X = cheap \cap Y = spam)}{P(Y = spam)}$$

$$= \underline{1}.$$

For Not Spam:

$$P(X = buy \mid Y = Not\ Spam) = \frac{P(X = buy \cap Y = not\ spam)}{P(Y = not\ spam)}$$

$$= \frac{1/4}{2/4}$$

$$= 1/2.$$

$P(X = cheap \mid Y = not\ spam)$

$$\frac{P(X = cheap \cap Y = Not\ spam)}{P(Y = not\ spam)} = \frac{1/4}{2/4} = 1/2$$

(b)

P(

(b) To compute the posterior probabilities.

Given that : X = cheap, not buy

First checking spam: we just need to check the numerator as denominator is same for both

$P(Y=1 \mid X=\text{cheap, not buy}) = P(Y=\text{spam}) \times P(X=\text{cheap} \mid Y=\text{spam})$

$\times P(X=\text{not buy} \mid Y=\text{spam})$

$= \frac{1}{2} \times \frac{1}{2} \times (1-1) = 0.$

Checking not spam:

$P(Y=\text{Not spam} \mid X=\text{cheap, Not buy})$

$= P(Y=\text{not spam}) \times P(X=\text{cheap} \mid Y=\text{not spam})$

$\times P(X=\text{not buy} \mid Y=\text{not spam})$

$= \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2}$

$= 1/8$

(c) The prob with prob. is that if a conditional prob is zero, the entire expression becomes zero showing lack of generalization and a complete aversion to unseen data.

To approaches to solve this is to use

m-estimate = c : numbers of classes = $P(A_i \mid C) = \frac{N_{ic}+mp}{N_c+m}$.

$P(A_i \mid C) = \frac{N_{ic}+1}{N_c+c}$ ← Laplace estimate

n_{ic} is the no. of times A_i and c occurs and n_c is the no. of times c occurs.

SECTION B

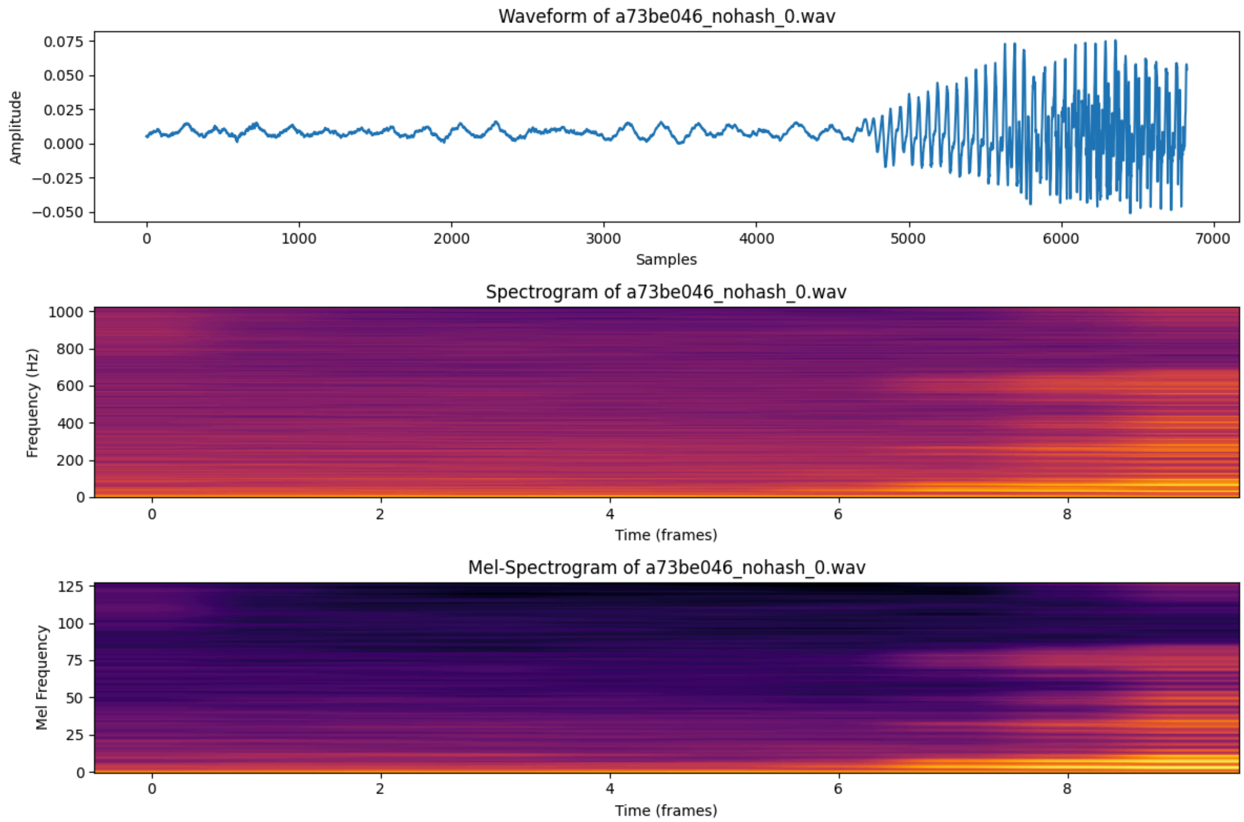
1 (a). Statistical summary of the data is given below:

	Mean				
	mean	median	min	max	std
Class					
_background_noise_	-0.004955	-1.775734e-05	-0.029636	-0.000007	0.012091
backward	0.000144	-2.651215e-07	-0.077356	0.088151	0.004145
bed	0.000097	-2.174734e-07	-0.077014	0.088236	0.006772
bird	-0.000143	3.795624e-07	-0.230286	0.088332	0.010779
cat	-0.000265	-2.574921e-07	-0.211095	0.087865	0.011951
dog	-0.000425	-1.331314e-07	-0.230329	0.088878	0.012447
down	0.000050	-3.643036e-07	-0.208074	0.088093	0.006024
eight	-0.000052	-3.471374e-07	-0.208647	0.331275	0.008099
five	-0.000010	2.784729e-07	-0.230202	0.088356	0.007253
follow	-0.000194	2.185822e-06	-0.157871	0.017067	0.005703
forward	0.000120	4.287720e-06	-0.009457	0.088124	0.002684
four	-0.000172	-3.814697e-08	-0.230293	0.062128	0.007559
go	-0.000184	1.323799e-08	-0.160621	0.064560	0.005164
happy	-0.000244	1.029968e-07	-0.230300	0.085434	0.011628
house	0.000146	-7.057190e-08	-0.155956	0.085782	0.005475
learn	-0.000004	1.346588e-06	-0.155392	0.087803	0.004846
left	0.000048	-2.632141e-07	-0.210939	0.088600	0.006803
marvin	-0.000342	4.005432e-07	-0.230317	0.085572	0.010735
nine	-0.000002	-6.961823e-07	-0.076602	0.088251	0.003824
no	0.000019	2.250671e-07	-0.230324	0.084674	0.006008
off	0.000044	-8.392334e-08	-0.119272	0.084558	0.004421
on	0.000047	-3.523421e-07	-0.209007	0.088544	0.005009
...					
yes	1.000000	0.067037	4044		
zero	1.000000	0.057204	4052		

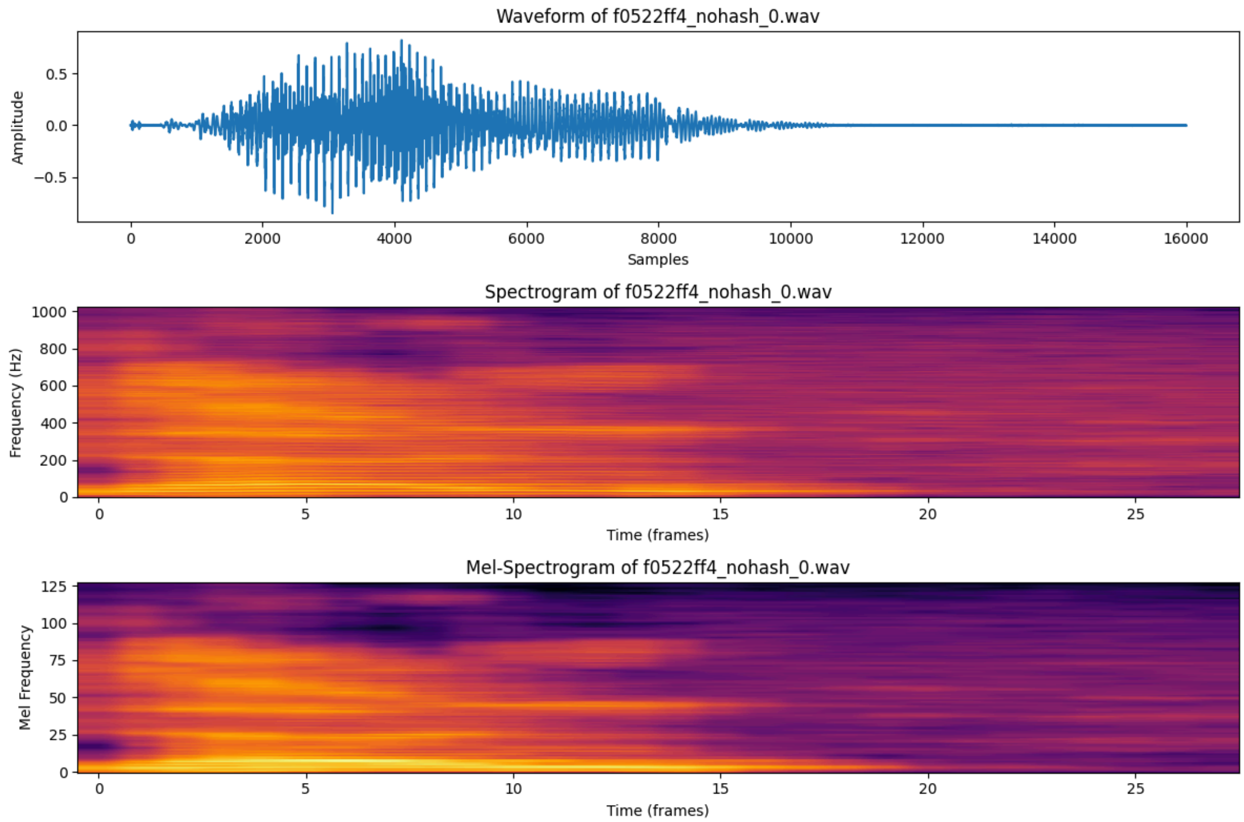
The mean amplitude is close to zero, which implies the audio has been normalized.

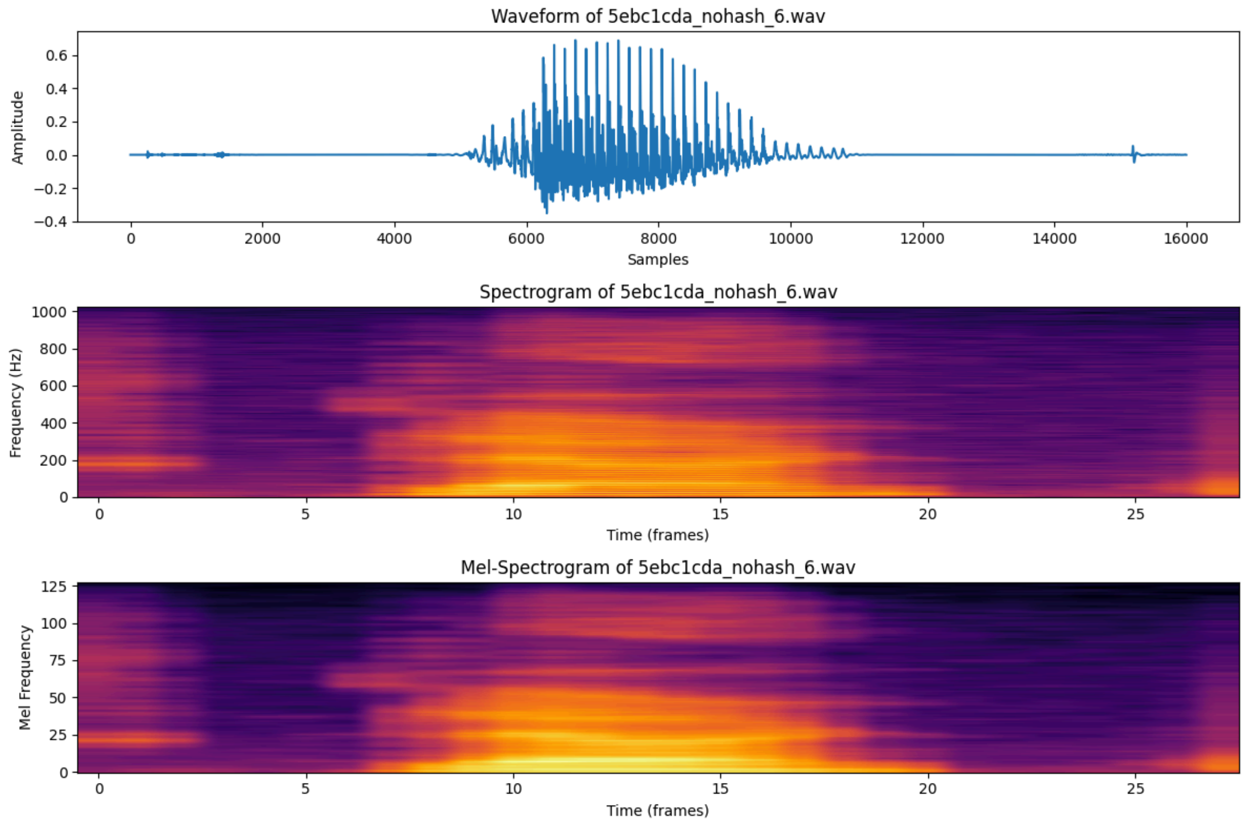
The std dev is quite low indicating the data is quite consistent.

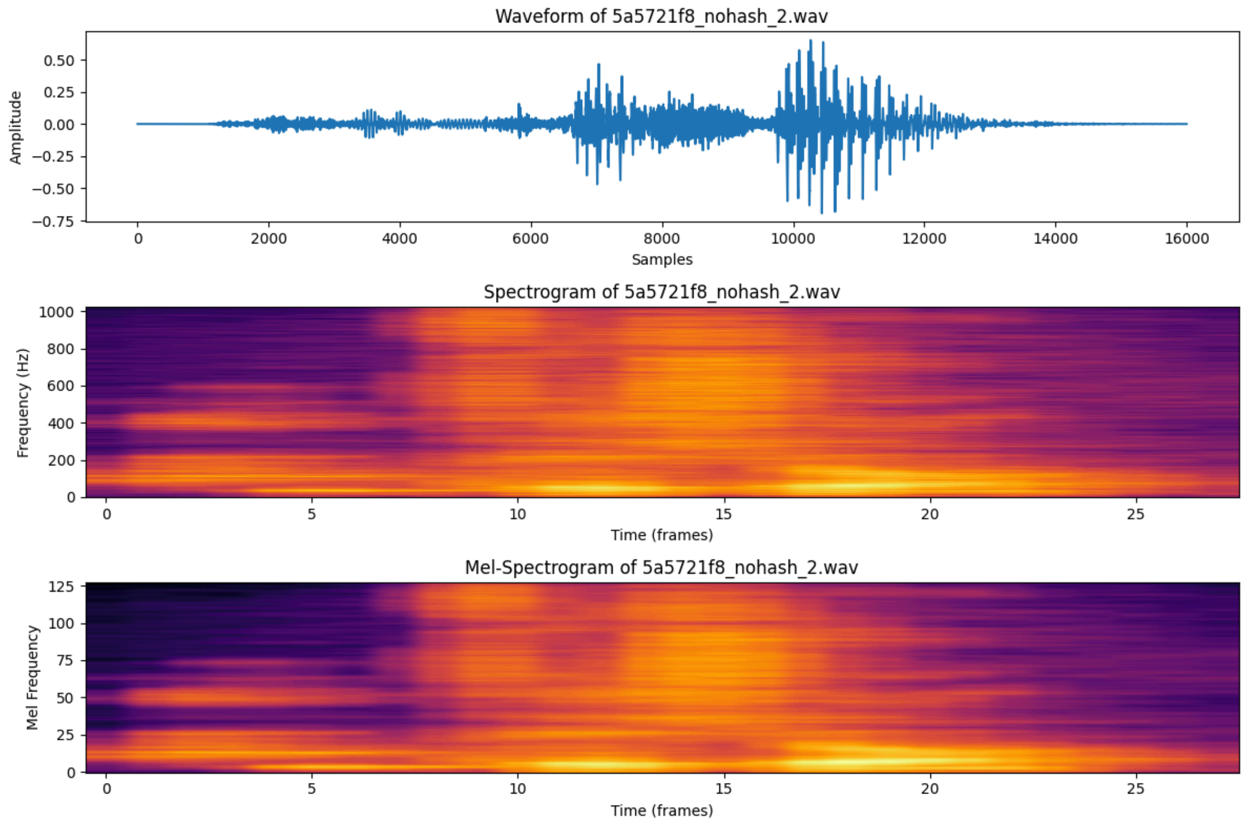
1 (b). Plots

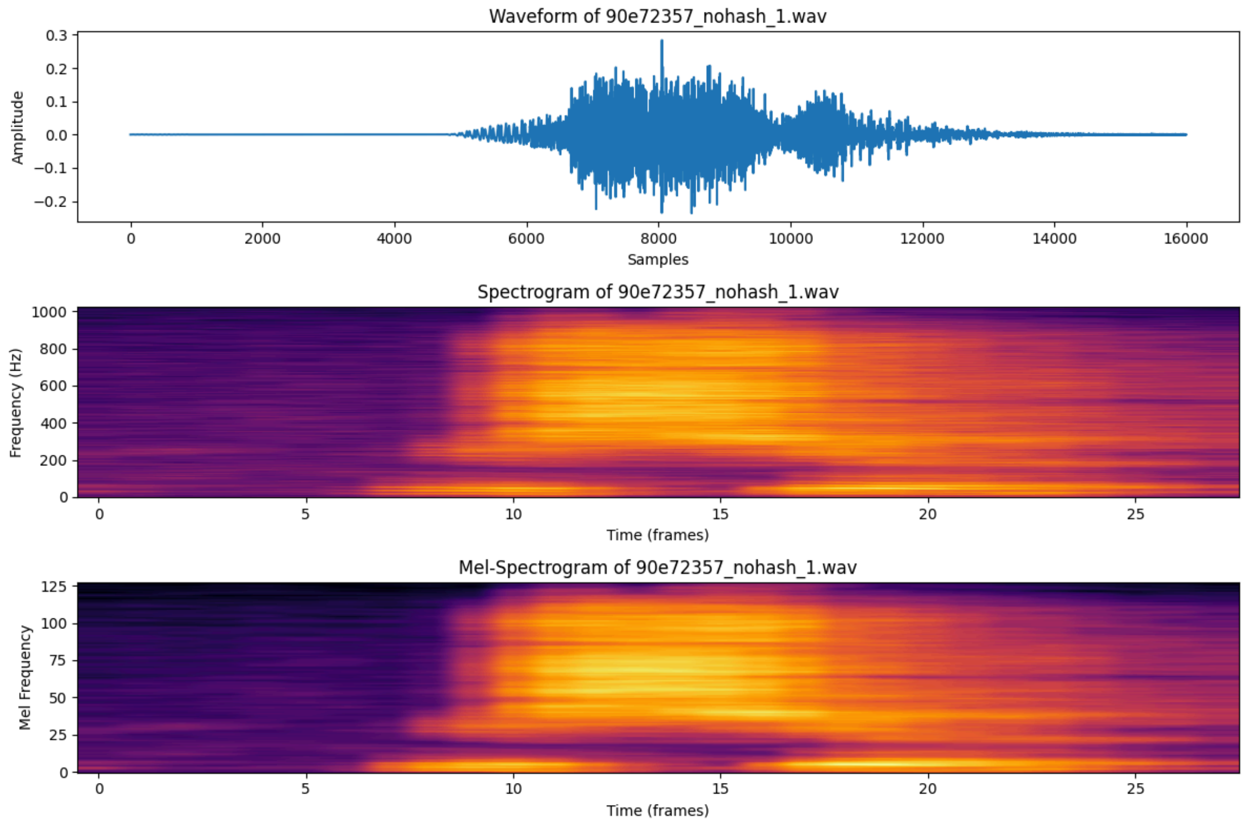


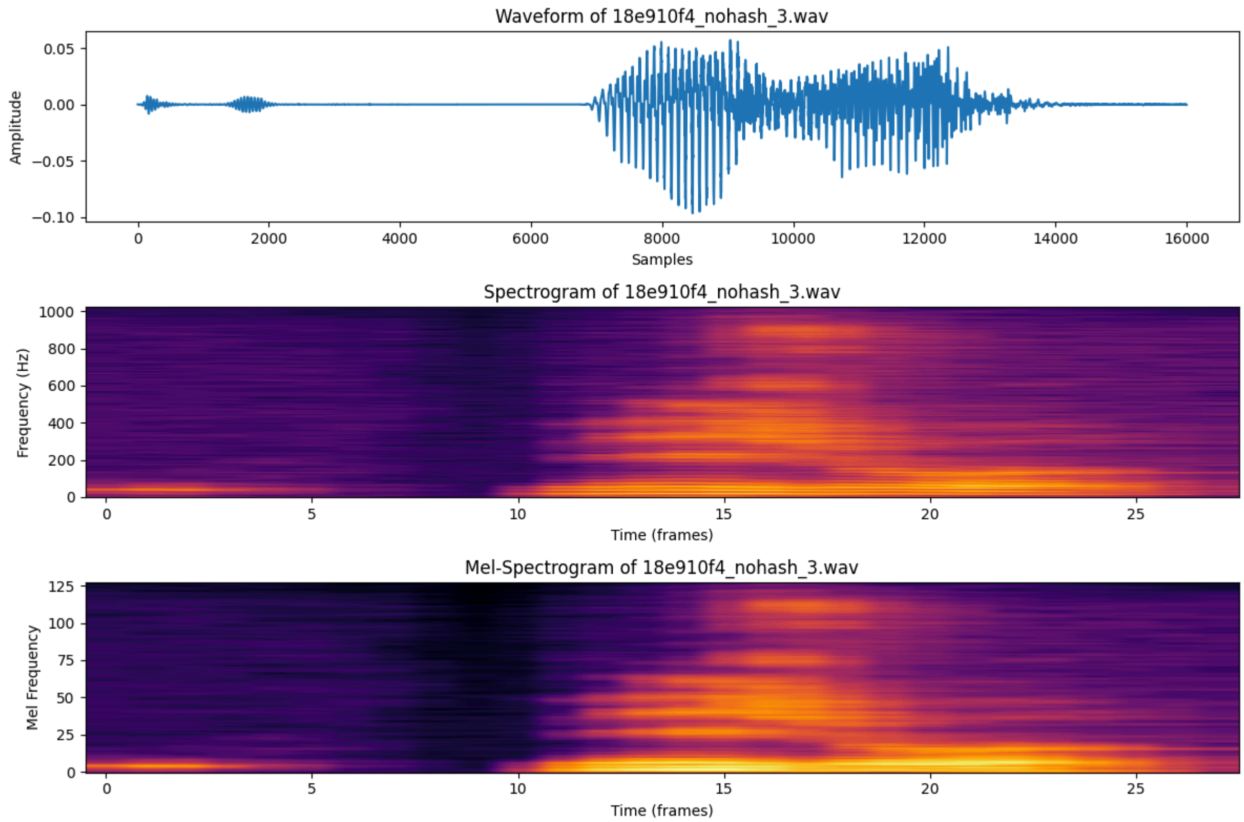
Waveform of a73be046_nohash_0.wav
Amplitude
0.075
0.050
0.025
0.000
−0.025
−0.050
0
1000
2000
3000
4000
5000
6000
7000
Samples
Spectrogram of a73be046_nohash_0.wav
Frequency (Hz)
1000
800
600
400
200
0
0
2
4
6
8
Time (frames)
Mel-Spectrogram of a73be046_nohash_0.wav
Mel Frequency
125
100
75
50
25
0
0
2
4
6
8
Time (frames)

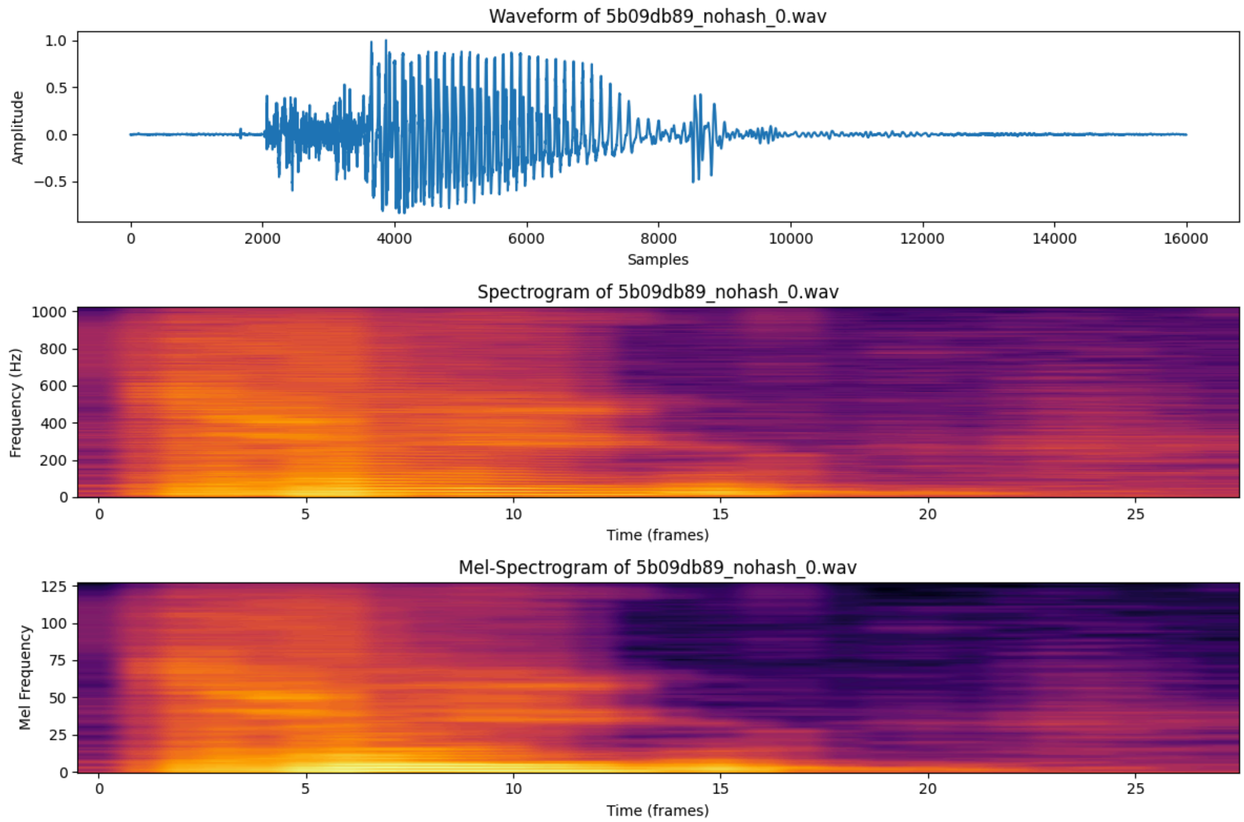

Waveform of f0522ff4_nohash_0.wav
Amplitude
0.5
0.0
−0.5
0
2000
4000
6000
8000
10000
12000
14000
16000
Samples
Spectrogram of f0522ff4_nohash_0.wav
Frequency (Hz)
1000
800
600
400
200
0
0
5
10
15
20
25
Time (frames)
Mel-Spectrogram of f0522ff4_nohash_0.wav
Mel Frequency
125
100
75
50
25
0
0
5
10
15
20
25
Time (frames)


Waveform of 5ebc1cda_nohash_6.wav
Amplitude
Samples
Spectrogram of 5ebc1cda_nohash_6.wav
Frequency (Hz)
Time (frames)
Mel-Spectrogram of 5ebc1cda_nohash_6.wav
Mel Frequency
Time (frames)

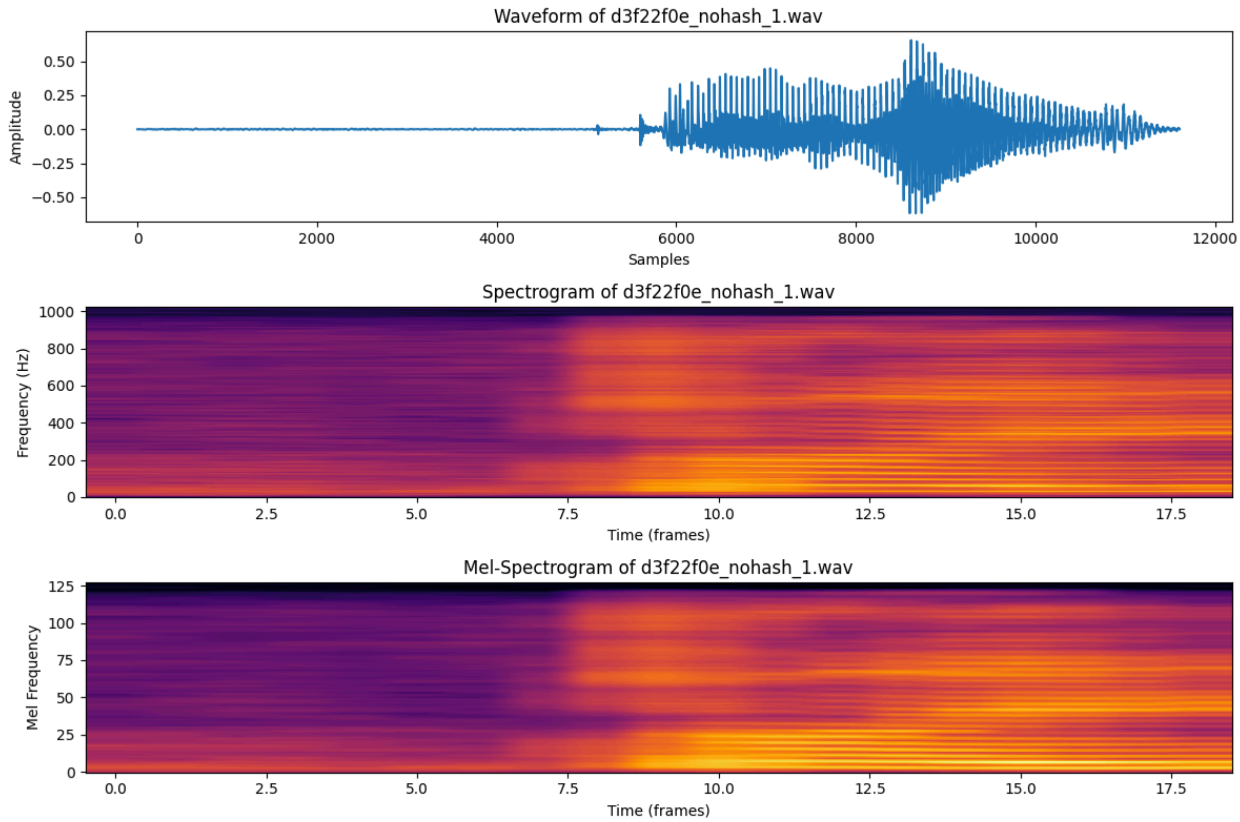

Waveform of 5a5721f8_nohash_2.wav
Amplitude
Samples
Spectrogram of 5a5721f8_nohash_2.wav
Frequency (Hz)
Time (frames)
Mel-Spectrogram of 5a5721f8_nohash_2.wav
Mel Frequency
Time (frames)

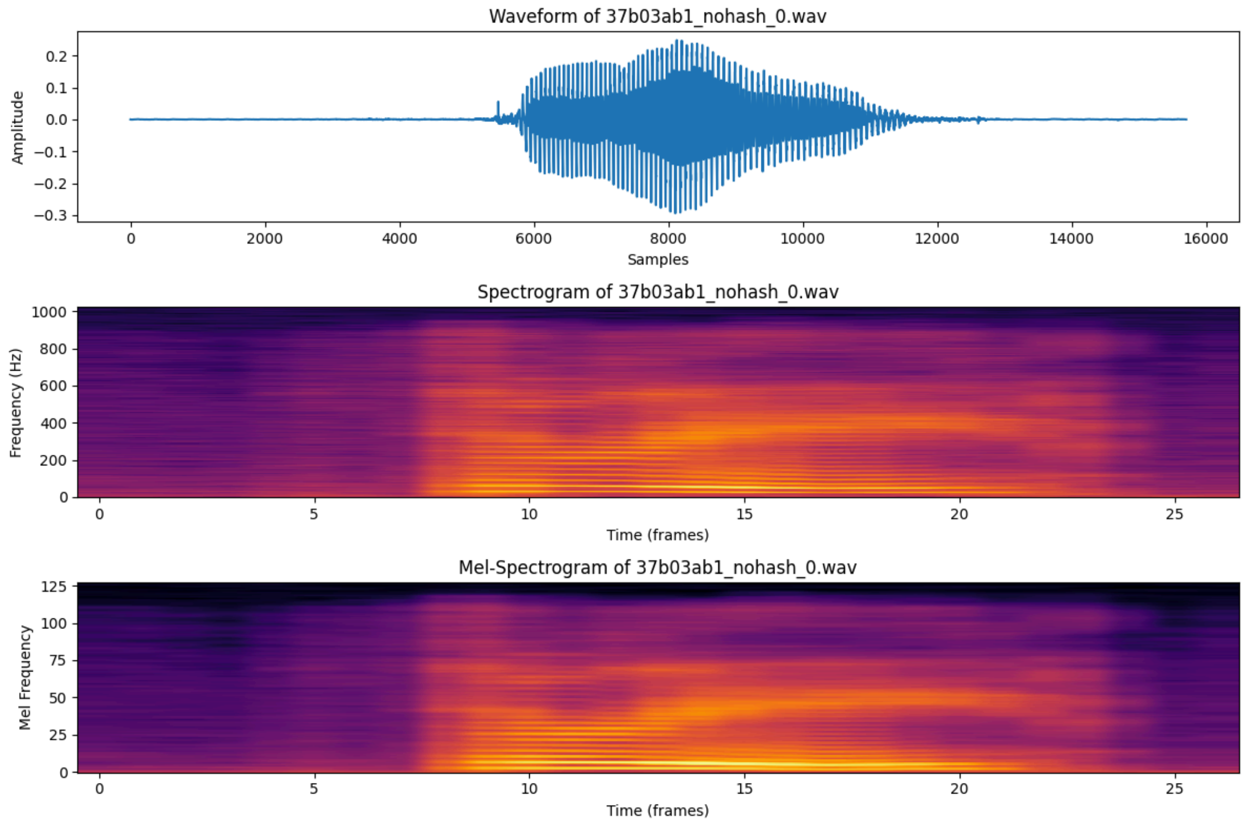

Waveform of 90e72357_nohash_1.wav
Amplitude
0.3
0.2
0.1
0.0
−0.1
−0.2
0
2000
4000
6000
8000
10000
12000
14000
16000
Samples
Spectrogram of 90e72357_nohash_1.wav
Frequency (Hz)
1000
800
600
400
200
0
0
5
10
15
20
25
Time (frames)
Mel-Spectrogram of 90e72357_nohash_1.wav
Mel Frequency
125
100
75
50
25
0
0
5
10
15
20
25
Time (frames)


Waveform of 18e910f4_nohash_3.wav
Amplitude
Samples
Spectrogram of 18e910f4_nohash_3.wav
Frequency (Hz)
Time (frames)
Mel-Spectrogram of 18e910f4_nohash_3.wav
Mel Frequency
Time (frames)



Waveform of 5b09db89_nohash_0.wav
Amplitude
1.0
0.5
0.0
−0.5
0
2000
4000
6000
8000
10000
12000
14000
16000
Samples
Spectrogram of 5b09db89_nohash_0.wav
Frequency (Hz)
1000
800
600
400
200
0
5
10
15
20
25
Time (frames)
Mel-Spectrogram of 5b09db89_nohash_0.wav
Mel Frequency
125
100
75
50
25
0
Time (frames)


Waveform of d3f22f0e_nohash_1.wav
Amplitude
Samples
Spectrogram of d3f22f0e_nohash_1.wav
Frequency (Hz)
Time (frames)
Mel-Spectrogram of d3f22f0e_nohash_1.wav
Mel Frequency
Time (frames)



Observations:

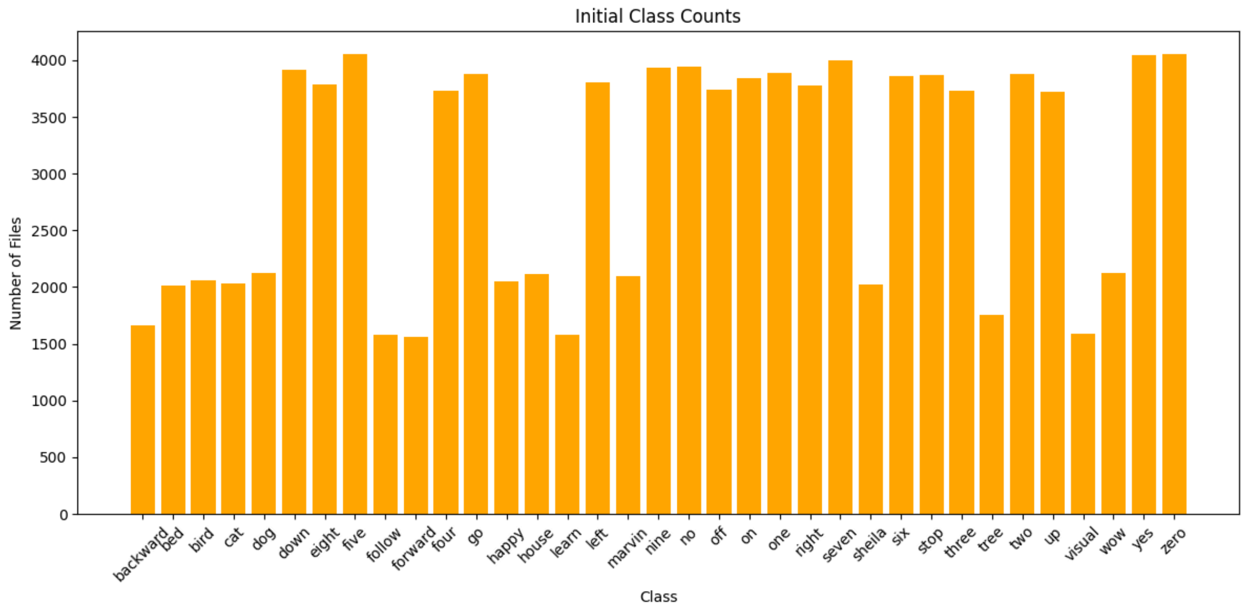
The amplitude fluctuates over time.

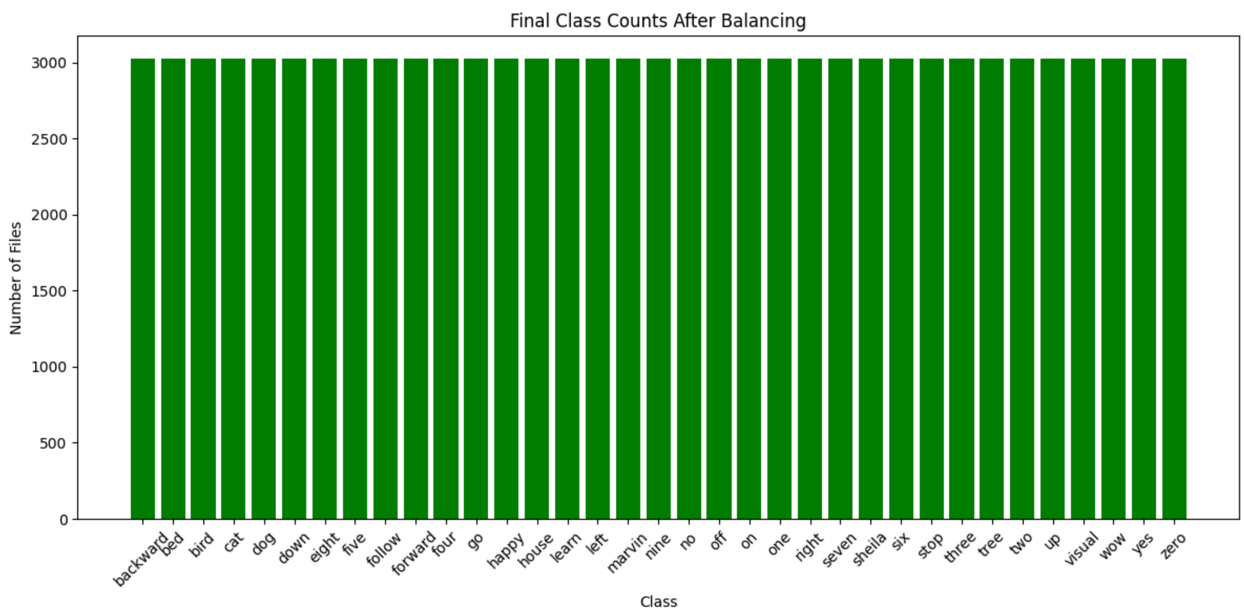
The waveform has a clear periodic pattern.

The frequency is considerably high. The frequency content appears to vary slightly over time, possibly indicating a changing pitch or timbre.

The frequency content appears to vary slightly over time, possibly indicating a changing pitch or timbre.

1 (c).Imbalances
There were imbalances and this was dealt with by using undersampling and oversampling all classes to avg number of files in a class.





1 (d). Data cleaning was done by removing silent segments.

2. Feature extraction

Features used:

MFCCs capture the short-term power spectrum of sound and are thus useful in speech recognition and classification.

Spectral Contrast measures the difference in amplitude between peaks and valleys in the sound spectrum, reflecting the timbral texture of the audio.

Spectral Rolloff is the frequency below which a certain percentage of the total spectral energy lies, often used to distinguish between harmonic and non-harmonic content.

Spectral Centroid is the frequency below which a certain percentage of the total spectral energy lies, often used to distinguish between harmonic and non-harmonic content.

Zero crossing rate is the rate at which the signal changes sign, indicating how often the audio signal crosses the zero amplitude level.

Root Mean Square energy provides a measure of the audio signal's power, reflecting its loudness over time.

3. Model

Random Forest performed best with 52.86% accuracy.

Hyperparameter tuning helped improve accuracy. The parameters were tried and tested by trial and error. The best performing parameters were:

n_estimators = 200

max_depth = 25

min_samples_split = 2

Naive Bayes performed poorly with Accuracy: 26.78%.
Perceptron was not used as it is only useful for binary classification.

SECTION C

1.(a) the number of images per class, the distribution of image sizes

Number of images per class:

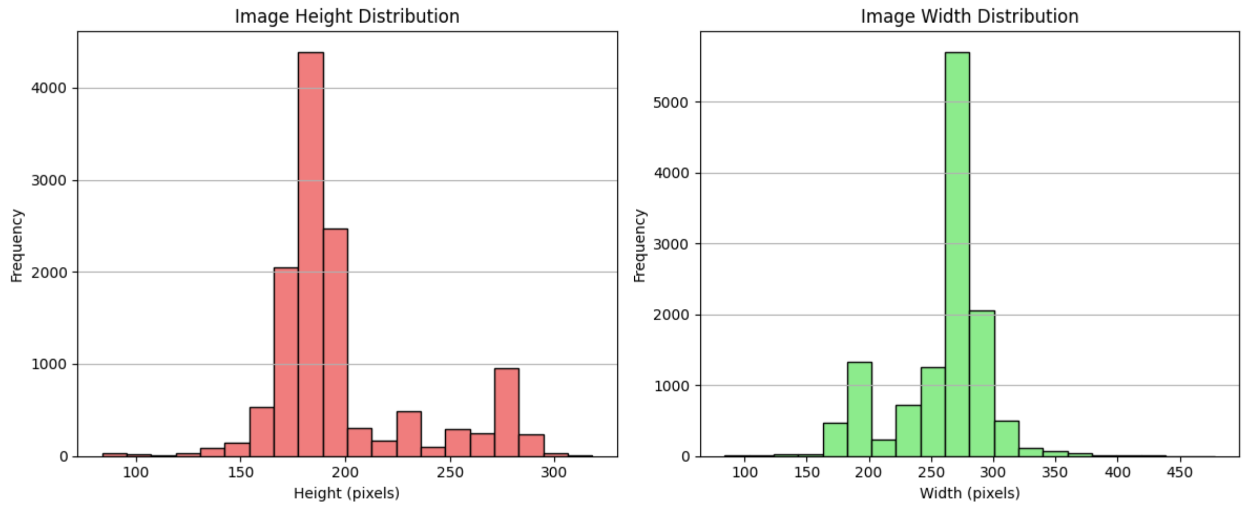
label

sitting 840

```
using_laptop        840
hugging             840
sleeping            840
drinking            840
clapping            840
dancing             840
cycling             840
calling             840
laughing            840
eating              840
fighting            840
listening_to_music  840
running             840
texting             840
Name: count, dtype: int64
```

Image size distribution:

	Height	Width
count	12601.000000	12601.000000
mean	196.574399	260.379652
std	35.280124	39.917997
min	84.000000	84.000000
25%	181.000000	254.000000
50%	183.000000	275.000000
75%	194.000000	276.000000
max	318.000000	478.000000



1.(b)

1 . (c)

Class distribution imbalance:

label

sitting 840

using_laptop 840

hugging 840

sleeping 840

drinking 840

clapping 840

dancing 840

cycling 840

calling 840

laughing 840

eating 840

fighting 840

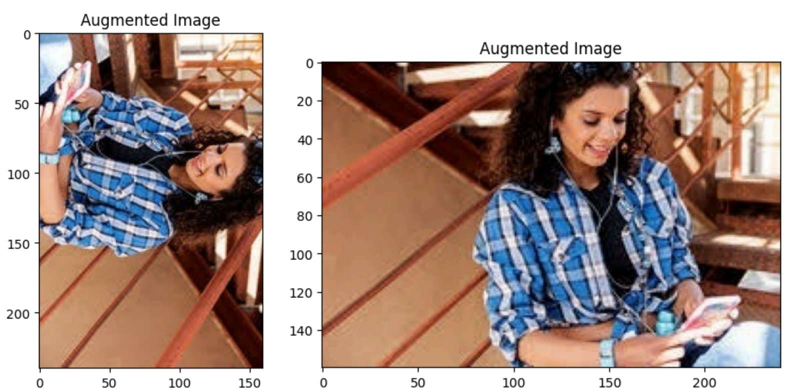
listening_to_music 840

running 840

texting 840

Name: count, dtype: int64

All classes are balanced with 840 images.



Data augmentation was not required as classes were balanced.

2. Feature Extraction:

Used features:

HOG - captures the distributions of localized gradients by computing a histogram using pixel intensities. Helps in object detection by using structural information.

Color histogram - captures the distribution of colors in an image by converting it to HSV color space and calculating histograms of associated values. Helps in object detection.

LBP - Local binary patterns help in feature extractions by using differences in localized pixel intensities. This helps in texture classification and facial recognition.

Gabor features - captures information about spatial structure and orientation of images, helping with texture analysis and edge detection.

GLCM - Gray Level Co-occurrence Matrix measures the frequency of pixel intensity combinations and calculates properties like contrast, energy, and homogeneity. It is useful in texture analysis and medical image classification

ORB (Oriented FAST and Rotated BRIEF) detects and describes local features like corners and edges. It is particularly useful in object recognition as it can deal with rotations, etc.

Other features I tried but were not useful:

Canny Edges - did not improve the feature set. Probably because Gabor already provides that functionality.

Hessian Matrix - This was very computationally intensive and didn't contribute to my accuracy.

3. Model Selection and Implementation

Random forest model was the best performing model. The highest accuracy recorded was 36.27 %.

Naive Bayes did not perform well with a low accuracy of 18.73%

Hyper parameter tuning helped improve accuracy of Random Forest Model. This was tested using GridSearchCV and also trial and error. Calculated estimates were based on the references below.
The accuracies for each set of hyperparameters were saved to a txt file.

hyperparameters = {'random_state':42, 'max_depth':16, 'n_estimators':7000, 'n_jobs':-1, 'min_samples_split':5, 'min_samples_leaf':2}

Perceptron was not trained as it is only useful for binary classification.

This was the hyperparameter set that performed the best.

REFERENCES:

Random forests - Machine Learning.

RandomForestClassifier — scikit-learn 1.5.2 documentation

Depth of Trees