

# 음성분석을 통한 감정분류

안장현, 황윤상, 장연철, 김명중, 김태효, 강읍, 박대원

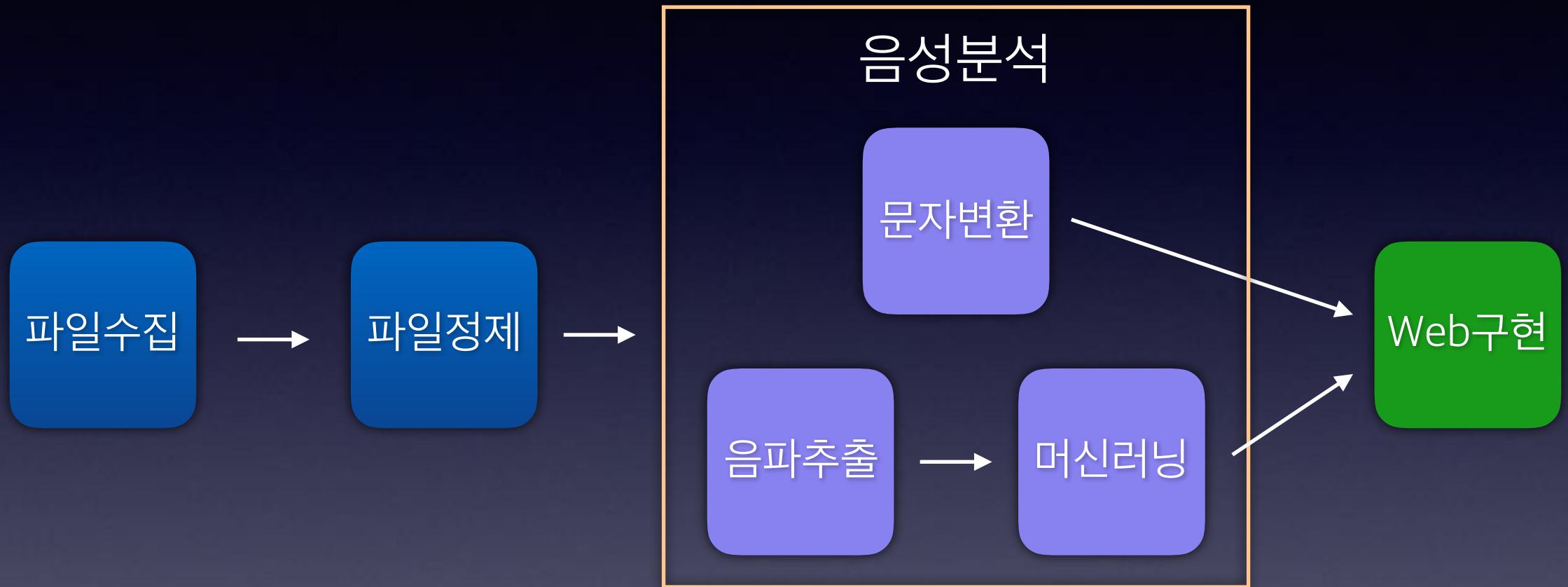


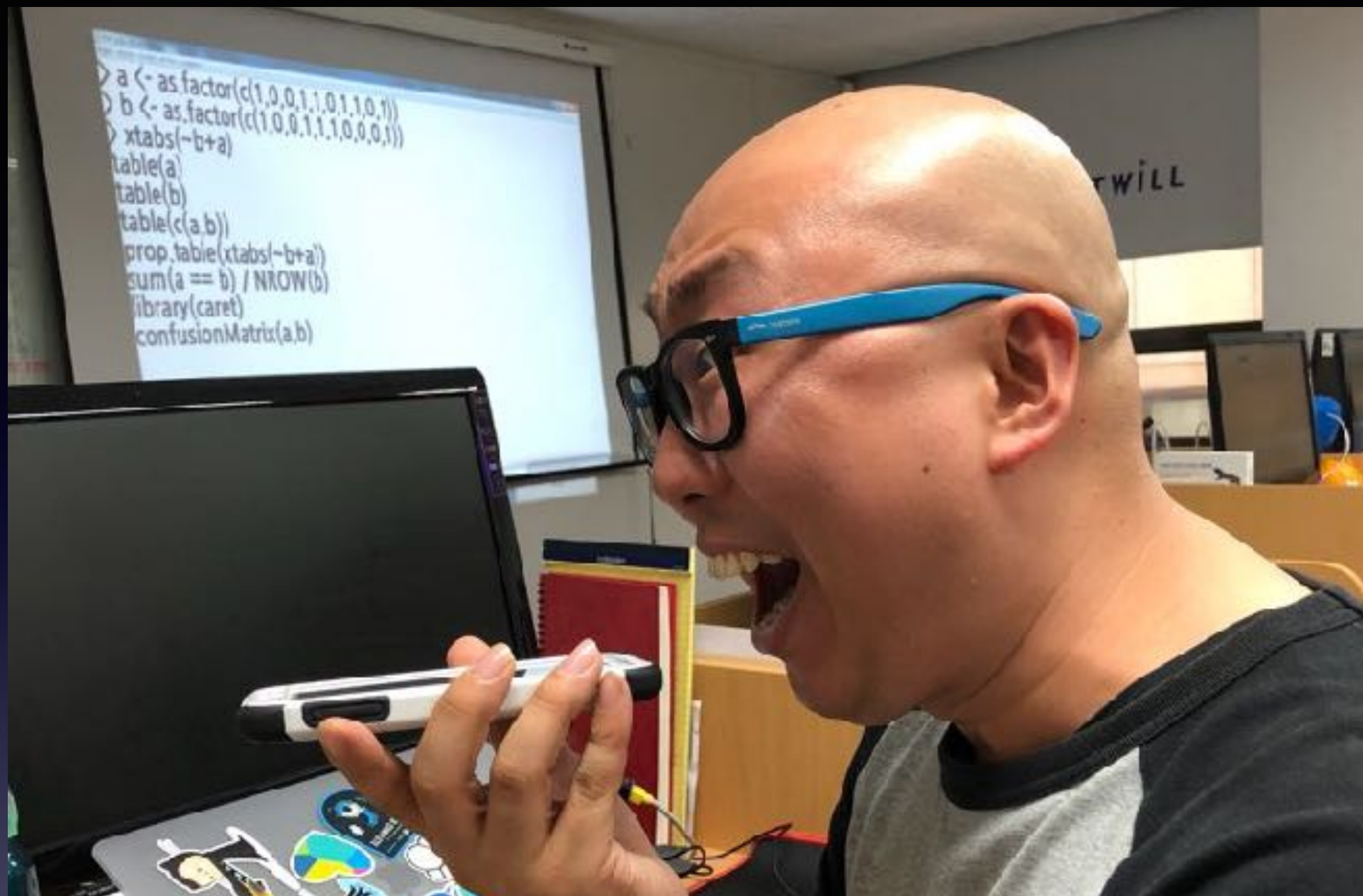
어떤 감정인지 궁금하지 않나요?

# 목 차

1. 작업순서
2. 데이터 수집 / 정제
3. 음성을 문자로 변환
4. 음파추출 / 가공
5. 머신 러닝
6. Web 으로 구현

# 1. 작업순서



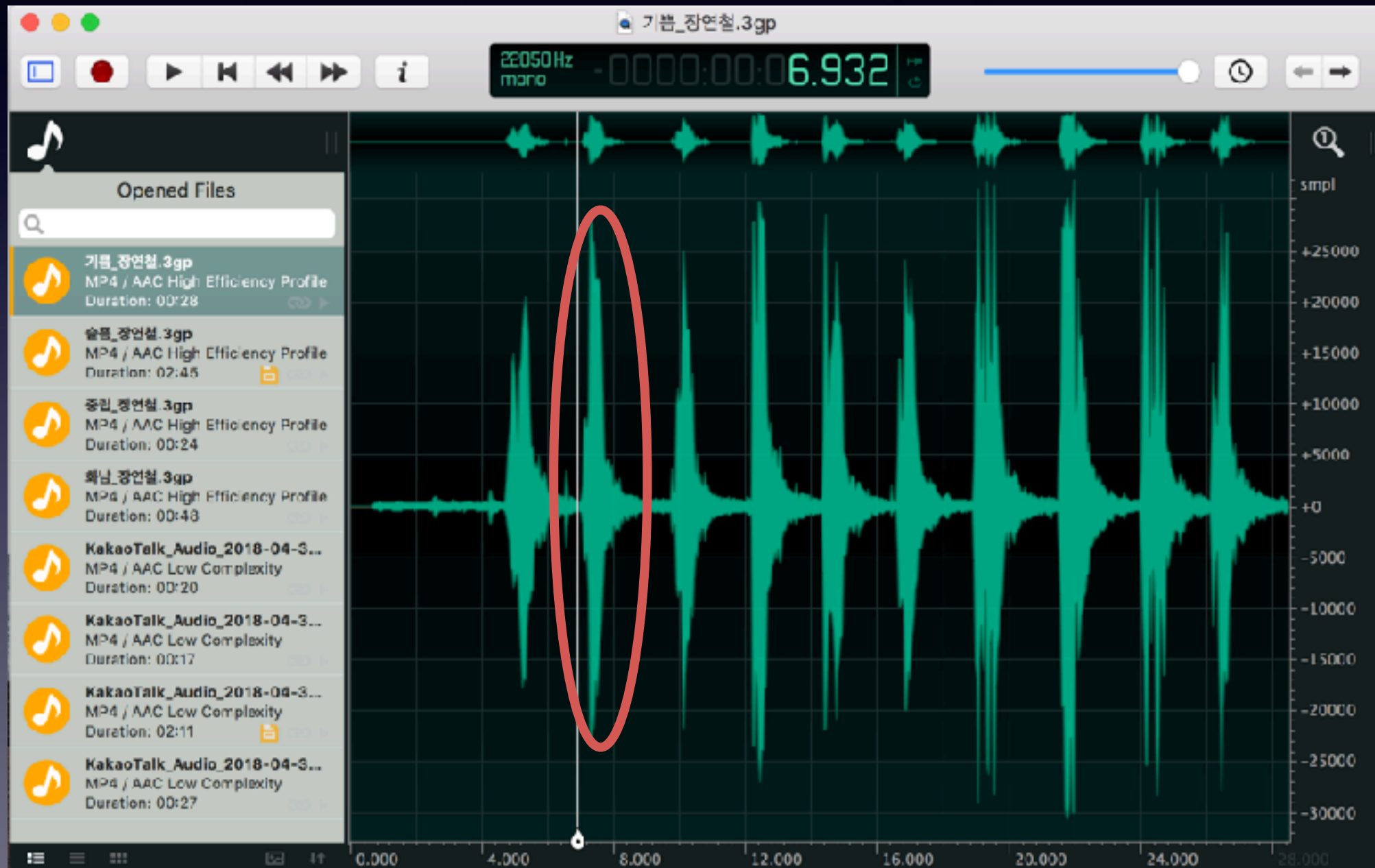


이렇게 직접 녹음!!

## 2. 데이터 수집 / 정제

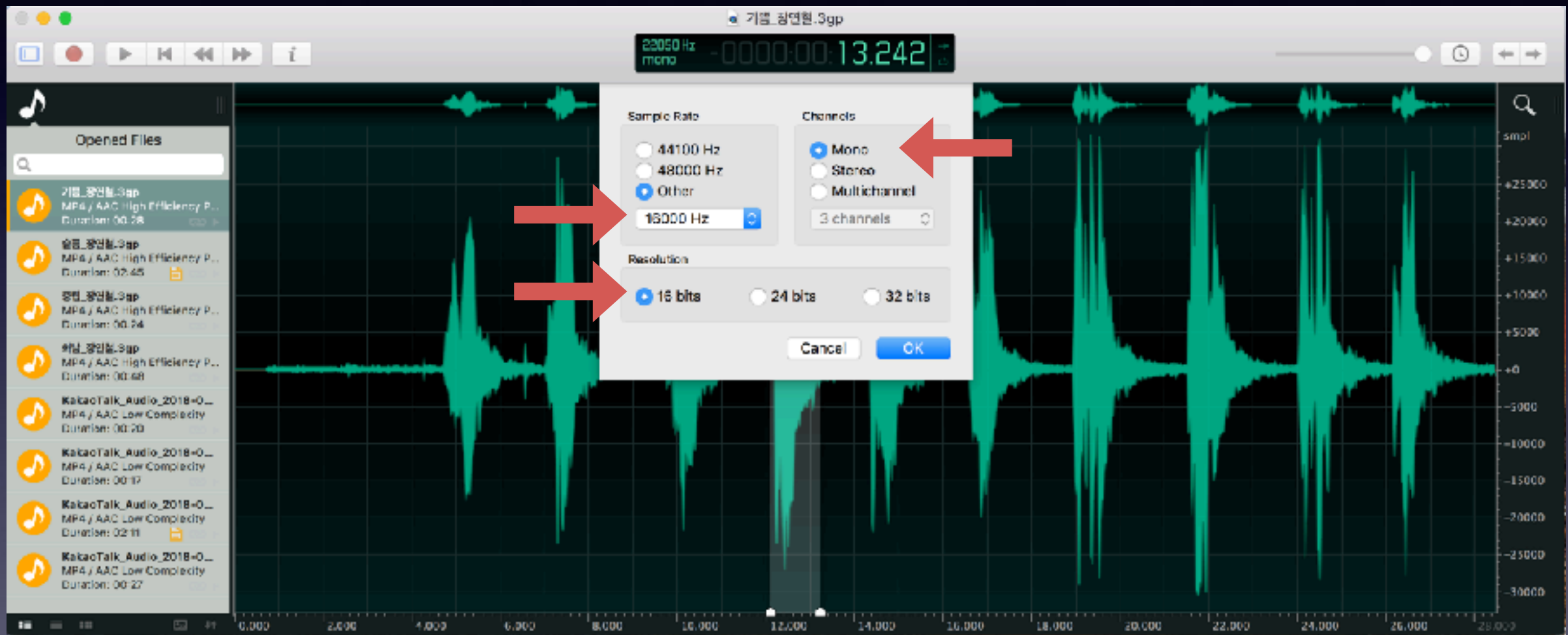


# 단어별 구간 분리

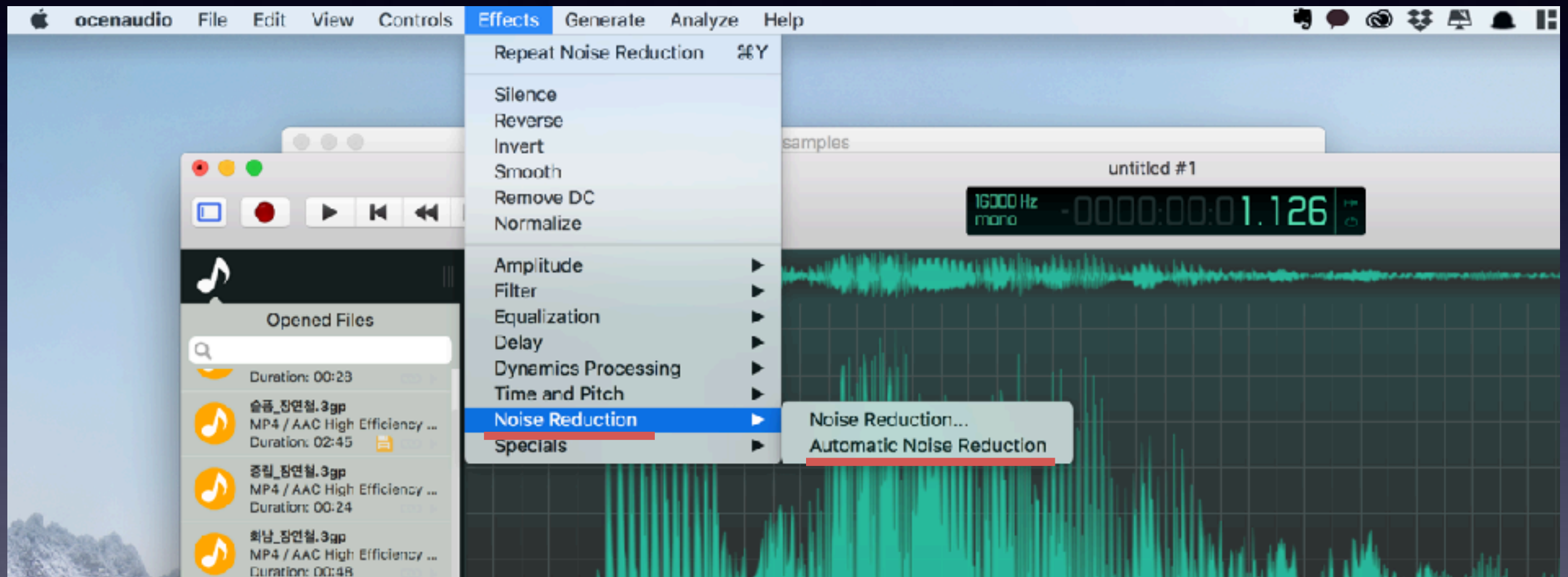




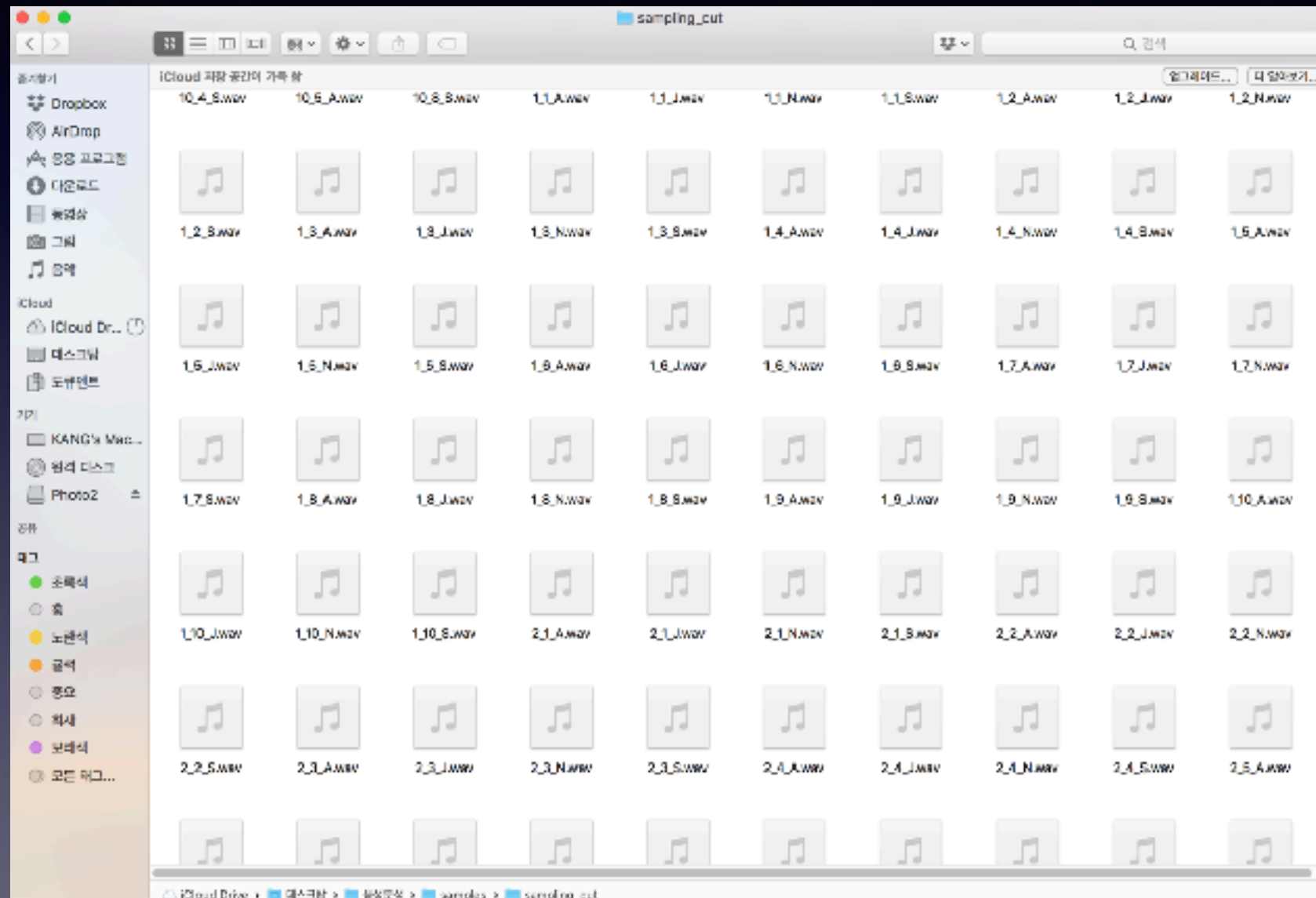
# 파일 설정



# 보정 작업



# 데이터 준비완료



307개

# 샘플링 분류

녹음한 내용

1 - “안녕하세요”

2 - “괜찮아요”

...

1\_1\_N.wav

녹음한 사람

감정 분류

N - Neutral

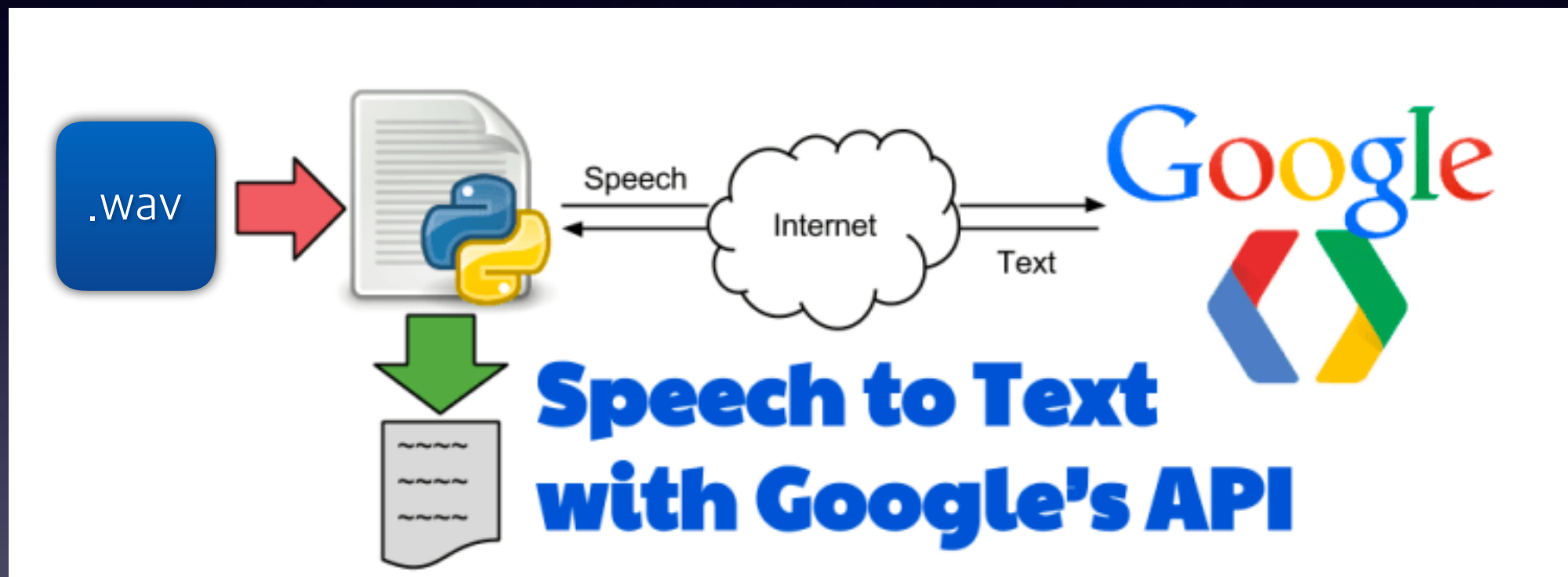
J - Joy

S - Sad

A - Angry

### 3. 음성을 문자로 변환

# 변환과정

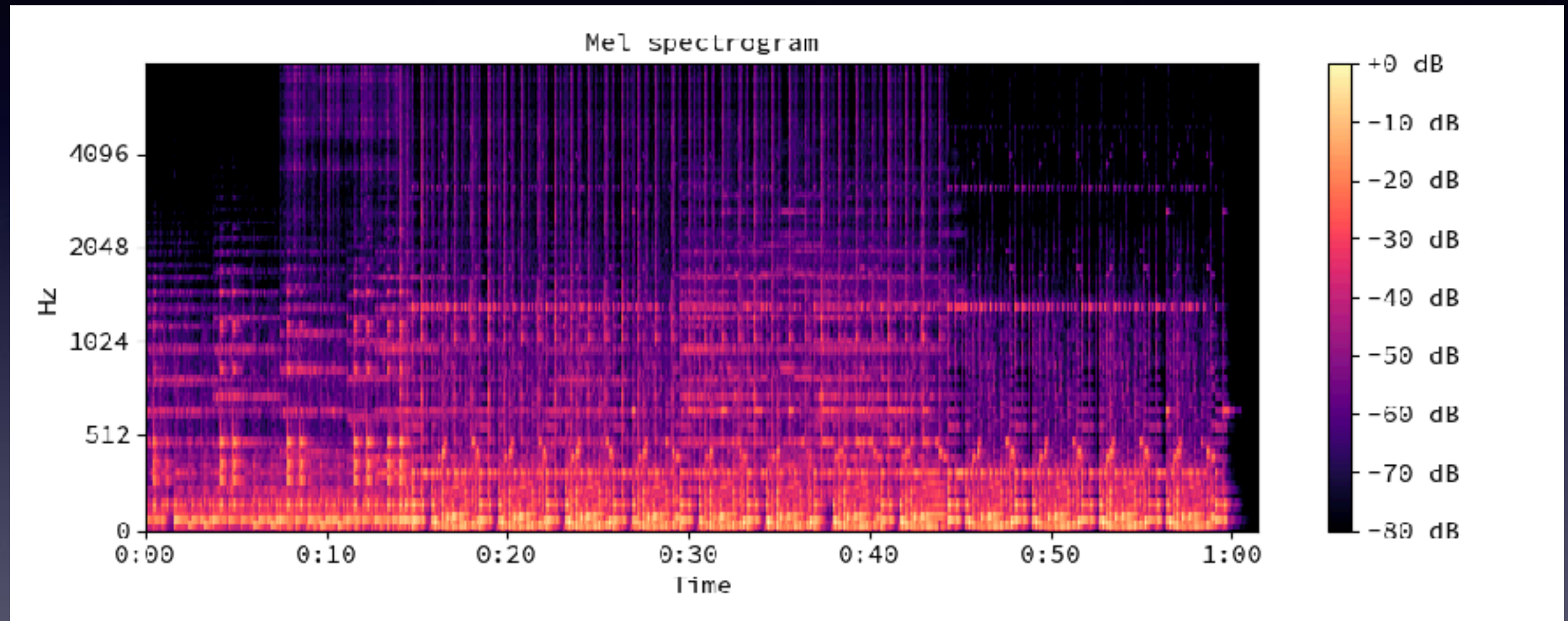


- Google Speech API 를 연동해서 구현

## 4. 음파추출 / 가공



# Librosa\_Mel Spectrogram



- 음성분석에 주로 사용

# 추출 방법

```
>>> import librosa
```

```
# 파일경로로 불러옴(채널 : mono, 샘플률 : 16000)
```

```
>>> r, sr = librosa.load(file_path, mono=True, sr=16000)
```

```
# 128행 N열 행렬로 나눔
```

```
>>> mel = librosa.feature.melspectrogram(y=r, sr=sr)
```

# 가공 방법

# 열 평균 구함

```
>>> a = np.mean(mel, axis=1)
```

# 정규화

```
>>> from sklearn.preprocessing import normalize
```

```
>>> data = normalize(a.reshape(1,128))
```

```
array([[1.97187563e-02, 1.31204324e-01, 3.77012200e-01, 8.06735781e-01,
        2.34193845e-01, 9.90510819e-02, 1.17132000e-01, 2.32765630e-01,
        1.22449384e-01, 6.81519214e-02, 4.31145196e-02, 6.66549872e-02,
        3.98691149e-02, 1.27058300e-02, 2.51292076e-02, 7.65263737e-02,
        5.34206733e-02, 3.14263556e-02, 5.06529453e-02, 9.48081741e-02,
        4.78322965e-02, 2.31508256e-02, 2.77047737e-02, 3.05280610e-02,
        1.03476818e-02, 1.05612901e-02, 1.08961047e-02, 6.46538299e-03,
        4.59660867e-03, 8.27244783e-03, 8.03706268e-03, 1.18324052e-02,
        5.85954128e-03, 4.79302841e-03, 1.33987971e-02, 2.93794040e-02,
        1.19293229e-02, 8.26804479e-03, 4.91111447e-03, 8.02826983e-03,
        7.66028126e-03, 5.36647150e-03, 9.02841907e-03, 9.72583953e-03,
        2.43884445e-03, 2.82264687e-03, 1.61121157e-03, 4.05665852e-03,
        4.36216770e-03, 8.46944875e-03, 6.09709487e-03, 5.66938415e-03,
        4.36060819e-03, 5.55885948e-03, 6.78262961e-03, 8.09273437e-03,
        5.95226058e-03, 2.62615179e-03, 2.46787021e-03, 3.02599866e-03,
        2.20879435e-03, 1.16494811e-03, 1.15627547e-03, 1.28382833e-03,
        1.25751848e-03, 2.88735103e-03, 1.28105714e-03, 7.42445844e-04,
        1.12612235e-03, 5.37626561e-04, 3.01923964e-04, 2.88441453e-04,
        3.66971806e-04, 4.63032743e-04, 7.40672525e-04, 1.34695724e-03,
        1.06598760e-03, 6.06939047e-04, 5.58377744e-04, 4.05844471e-04,
        2.65732180e-04, 1.44953430e-04, 1.52969060e-04, 1.54640936e-04,
        1.01729290e-04, 5.45085840e-05, 3.50769137e-05, 2.79371354e-05,
        3.91128870e-05, 5.58438539e-05, 7.01319647e-05, 4.41151258e-05,
        1.89070236e-05, 1.20011998e-05, 2.00269527e-05, 2.89060417e-05,
        3.32762000e-05, 2.04886112e-05, 2.42265499e-05, 3.54609879e-05,
```

# 5. 머신 러닝

# MLPClassifier



- sklearn 패키지에서 MLPClassifier 사용(다중분류시 사용)
- 라벨값은 슬픔(0), 중립(1), 기쁨(2), 화남(3)

# 모델 생성

```
>>> from sklearn.neural_network import MLPClassifier
```

```
>>> mlp_multilabel = MLPClassifier(  
    hidden_layer_sizes=(128,128), # 은닉층  
    max_iter=1000, # 최대 반복 횟수  
    random_state=None # 난수 생성(기본값)  
).fit(x_train, y_train) # 훈련데이터
```

```
>>> acc = mlp_multilabel.score(x_train, y_train) # 정확도
```



# 모델 검증

```
>>> y_pred = mlp_multilabel.predict(x_test)
```

```
# 검증 : (맞은 수) / (전체 수)
```

```
>>> print('모델 검증 :', np.sum(np.sum(y_test.astype(int)  
    & y_pred, axis=1) > 0)/y_test.shape[0])
```

# 모델 적용

# 생성된 모델에 의한 예측값

```
>>> pred = mlp_multilabel.predict(DATA)
```

```
>>> for i in pred: # 예측값에 해당하는 감정 분류
    if i == 0:
        res = "슬픔"
    elif i == 1:
        res = "중립"
    elif i == 2:
        res = "기쁨"
    elif i == 3:
        res = "화남"
```

## 6. Web 으로 구현

# Flask 란?



- 파이썬 웹 어플리케이션을 만드는 프레임 워크 중 하나
- 매우 심플하고 가벼운 느낌이 특징

# Flask 실행(ex.hello world)

```
>>> from flask import Flask
```

```
>>> app = Flask(__name__)
```

```
>>> @app.route('/')
>>> def hello_world():
```

```
    return 'hello world!'
```

```
>>> if __name__ == '__main__':
    app.debug = True
    app.run()
```

# Flask 실행결과

A screenshot of a web browser's address bar. It features three navigation icons on the left: a back arrow, a forward arrow, and a circular refresh icon. To the right of these icons is a document icon followed by the text "127.0.0.1:5000".

← → ↻ 127.0.0.1:5000

Hello, Python!!

# 참고 논문

- 음성 신호를 사용한 감정인식의 특징 파라미터 비교(2003), 김원구
- 감성기반 서비스를 위한 통화 음성 감정인식 기법(2014), 방재훈 외 1명
- 음성신호 기반 감정인식을 위한 특징 파라미터 비교(2016), 김남균 외 3명
- 음성의 피치 파라미터를 사용한 감정 인식(2015), 이규현 외 1명
- 음성 신호를 사용한 GMM기반의 감정 인식(2004), 강면규 외 2명
- Librosa: Audio and Music Signal Analysis in Python(2015), Brian McFee



감사합니다

시연

<http://renardy.iptime.org>