

pythonProjects

v1

Generated by Doxygen 1.11.0

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

main	??
ses_tanima	
Ses Tanıma Projesi	??

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

C:/Users/orakn/Desktop/VoiceReco/pythonProject/[main.py](#) ??

Chapter 3

Namespace Documentation

3.1 main Namespace Reference

Functions

- [reduce_noise](#) ([audio](#), [sr](#))
Gürültü azaltma için bir fonksiyon.
- [extract_features](#) ([file_path](#))
Öznitelik çıkartmak için bir fonksiyon.
- [extract_features_from_audio](#) ([audio](#), [sample_rate](#))
- [plot_histogram](#) ([features](#), [speaker_label](#))
- [plot_mel_spectrogram](#) ([audio](#), [sr](#), [speaker_label](#))
- [transcribe_speech](#) ([audio](#), [sample_rate](#))
- [recognize_from_microphone](#) ()

Variables

- list [ses_dosyolari](#)
- list [konusmaci_etiketleri](#)
- list [X](#) = []
- list [y](#) = []
- [features](#)
- [audio](#)
- [sample_rate](#)
- [test_size](#) = max(0.2, 3 / len(y))
- [konuşmacı_indeksleri](#) = defaultdict(list)
- [X_train](#)
- [X_test](#)
- [y_train](#)
- [y_test](#)
- [train_size](#) = int(len(indeksler) * (1 - [test_size](#)))
- [train_indeksler](#) = indeksler[:[train_size](#)]
- [test_indeksler](#) = indeksler[[train_size](#):]
- [scaler](#) = StandardScaler()
- [X_train_scaled](#) = scaler.fit_transform([X_train](#))
- [X_test_scaled](#) = scaler.transform([X_test](#))
- [model](#) = SVC(kernel='linear')

- `train_accuracy` = `model.score(X_train_scaled, y_train)`
- `test_accuracy` = `model.score(X_test_scaled, y_test)`
- `y_pred` = `model.predict(X_test_scaled)`
- `accuracy` = `accuracy_score(y_test, y_pred)`
- `precision` = `precision_score(y_test, y_pred, average='macro', zero_division=0)`
- `recall` = `recall_score(y_test, y_pred, average='macro', zero_division=0)`
- `f1` = `f1_score(y_test, y_pred, average='macro', zero_division=0)`

3.1.1 Function Documentation

3.1.1.1 `extract_features()`

```
main.extract_features (
    file_path)
```

Öznitelik çıkartmak için bir fonksiyon.

Parameters

<code>file_path</code>	Ses dosyasının yolu
------------------------	---------------------

Returns

Öznitelik vektörü, ses verisi, örnekleme oranı

3.1.1.2 `extract_features_from_audio()`

```
main.extract_features_from_audio (
    audio,
    sample_rate)
```

3.1.1.3 `plot_histogram()`

```
main.plot_histogram (
    features,
    speaker_label)
```

3.1.1.4 `plot_mel_spectrogram()`

```
main.plot_mel_spectrogram (
    audio,
    sr,
    speaker_label)
```

3.1.1.5 `recognize_from_microphone()`

```
main.recognize_from_microphone ()
```

3.1.1.6 `reduce_noise()`

```
main.reduce_noise (
    audio,
    sr)
```

Gürültü azaltma için bir fonksiyon.

Parameters

<i>audio</i>	Ses verisi
<i>sr</i>	Örnekleme oranı

Returns

Gürültüsü azaltılmış ses verisi Gürültü azaltma için bir fonksiyon tanımlayalım

3.1.1.7 transcribe_speech()

```
main.transcribe_speech (  
    audio,  
    sample_rate)
```

3.1.2 Variable Documentation**3.1.2.1 accuracy**

```
main.accuracy = accuracy_score(y_test, y_pred)
```

3.1.2.2 audio

```
main.audio
```

3.1.2.3 f1

```
main.f1 = f1_score(y_test, y_pred, average='macro', zero_division=0)
```

3.1.2.4 features

```
main.features
```

3.1.2.5 konuşmacı_indeksleri

```
main.konuşmacı_indeksleri = defaultdict(list)
```

3.1.2.6 konusmaci_etiketleri

```
list main.konusmaci_etiketleri
```

Initial value:

```
00001 = [  
00002     "konusmaci1", "konusmaci1",  
00003     "konusmaci2", "konusmaci2",  
00004     "konusmaci3", "konusmaci3",  
00005     "konusmaci4", "konusmaci4"  
00006 ]
```

3.1.2.7 model

```
main.model = SVC(kernel='linear')
```

3.1.2.8 precision

```
main.precision = precision_score(y_test, y_pred, average='macro', zero_division=0)
```

3.1.2.9 recall

```
main.recall = recall_score(y_test, y_pred, average='macro', zero_division=0)
```

3.1.2.10 sample_rate

```
main.sample_rate
```

3.1.2.11 scaler

```
main.scaler = StandardScaler()
```

3.1.2.12 ses_dosyalari

```
list main.ses_dosyalari
```

Initial value:

```
00001 = [  
00002     "konusmaci1/nurullah_train.wav", "konusmaci1/nurullah_test.wav",  
00003     "konusmaci2/enes_train.wav", "konusmaci2/enes_test1.wav",  
00004     "konusmaci3/voice_yasin.wav", "konusmaci3/yasin_test1.wav",  
00005     "konusmaci4/halil_train.wav", "konusmaci4/halil_test.wav"  
00006 ]
```

3.1.2.13 test_accuracy

```
main.test_accuracy = model.score(X_test_scaled, y_test)
```

3.1.2.14 test_indeksler

```
main.test_indeksler = indeksler[train_size:]
```

3.1.2.15 test_size

```
main.test_size = max(0.2, 3 / len(y))
```

3.1.2.16 train_accuracy

```
main.train_accuracy = model.score(X_train_scaled, y_train)
```

3.1.2.17 train_indeksler

```
main.train_indeksler = indeksler[:train_size]
```

3.1.2.18 train_size

```
main.train_size = int(len(indeksler) * (1 - test_size))
```

3.1.2.19 X

```
list main.X = []
```

3.1.2.20 X_test

```
main.X_test
```

3.1.2.21 X_test_scaled

```
main.X_test_scaled = scaler.transform(X_test)
```

3.1.2.22 X_train

```
main.X_train
```

3.1.2.23 X_train_scaled

```
main.X_train_scaled = scaler.fit_transform(X_train)
```

3.1.2.24 y

```
list main.y = []
```

3.1.2.25 y_pred

```
main.y_pred = model.predict(X_test_scaled)
```

3.1.2.26 `y_test`

`main.y_test`

3.1.2.27 `y_train`

`main.y_train`

3.2 `ses_tanima` Namespace Reference

Ses Tanima Projesi.

3.2.1 Detailed Description

Ses Tanima Projesi.

Version

1.0

Date

2023

Chapter 4

File Documentation

4.1 C:/Users/orakn/Desktop/VoiceReco/pythonProject/main.py File Reference

Namespaces

- namespace [main](#)
- namespace [ses_tanima](#)

Ses Tanıma Projesi.

Functions

- [main.reduce_noise](#) ([audio](#), [sr](#))
Gürültü azaltma için bir fonksiyon.
- [main.extract_features](#) ([file_path](#))
Öznitelik çıkartmak için bir fonksiyon.
- [main.extract_features_from_audio](#) ([audio](#), [sample_rate](#))
- [main.plot_histogram](#) ([features](#), [speaker_label](#))
- [main.plot_mel_spectrogram](#) ([audio](#), [sr](#), [speaker_label](#))
- [main.transcribe_speech](#) ([audio](#), [sample_rate](#))
- [main.recognize_from_microphone](#) ()

Variables

- list [main.ses_dosyaları](#)
- list [main.konusmaci_etiketleri](#)
- list [main.X](#) = []
- list [main.y](#) = []
- [main.features](#)
- [main.audio](#)
- [main.sample_rate](#)
- [main.test_size](#) = max(0.2, 3 / len(y))
- [main.konusmaci_indeksleri](#) = defaultdict(list)
- [main.X_train](#)
- [main.X_test](#)
- [main.y_train](#)

- `main.y_test`
- `main.train_size = int(len(indeksler) * (1 - test_size))`
- `main.train_indeksler = indeksler[:train_size]`
- `main.test_indeksler = indeksler[train_size:]`
- `main.scaler = StandardScaler()`
- `main.X_train_scaled = scaler.fit_transform(X_train)`
- `main.X_test_scaled = scaler.transform(X_test)`
- `main.model = SVC(kernel='linear')`
- `main.train_accuracy = model.score(X_train_scaled, y_train)`
- `main.test_accuracy = model.score(X_test_scaled, y_test)`
- `main.y_pred = model.predict(X_test_scaled)`
- `main.accuracy = accuracy_score(y_test, y_pred)`
- `main.precision = precision_score(y_test, y_pred, average='macro', zero_division=0)`
- `main.recall = recall_score(y_test, y_pred, average='macro', zero_division=0)`
- `main.f1 = f1_score(y_test, y_pred, average='macro', zero_division=0)`