

옥타브 알고리즘 문제점

옥타브 유사도의 경우도 마찬가지로 싱크 문제가 있어
보임 -> 0.1초 차이로 노트의 값이 틀어졌을 경우 알고
리즘 계산과정에서 낮은 정확도를 얻을 수 밖에 없음.

옥타브 알고리즘 데이터 전처리 추가

	sec	msg_type	channel	note	velocity	dynamic	accent	count	main_vol	depth	pedal	pan
0	0.0					ppp	0	0	100	0	0	0
1	0.1	['note_on', 'note_on']	[0, 0]	[60, 60]	[21, 21]	ppp	0	2	0	0	0	0
2	0.2					NaN	0	0	0	0	0	0
3	0.3	['note_on', 'note_on']	[0, 0]	[64, 64]	[32, 32]	ppp	0	2	0	0	0	0
4	0.4					NaN	0	0	0	0	0	0



	sec	msg_type	channel	note	velocity	new_octave	dynamic	accent	count	main_vol	depth	pedal	pan
0	0.0					{}	ppp	0	0	100	0	0	0
1	0.1	['note_on', 'note_on']	[0, 0]	[60, 60]	[21, 21]	{4}	ppp	0	2	0	0	0	0
2	0.2					{4}	NaN	0	0	0	0	0	0
3	0.3	['note_on', 'note_on']	[0, 0]	[64, 64]	[32, 32]	{4}	ppp	0	2	0	0	0	0
4	0.4					{4}	NaN	0	0	0	0	0	0

midi note 에 따른 octave 값

Octave	MIDI Note Range
-1	0-11
0	12-23
1	24-35
2	36-47
3	48-59
4	60-71
5	72-83
6	84-95
7	96-107
8	108-119
9	120-127

행간 옥타브 평가를 위해서 new_octave 열을 추가함.

옥타브 알고리즘 수정

new_octave에 따라서 octave 값을 단순 행간 비교

싱크 문제를 해결하기 위해 threshold 매개변수를 추가해, target_data를 기준으로 input_data의 위 아래 행과 비교해 값이 일치하다면, 맞는 것으로 판단함.

Octave	MIDI Note Range
-1	0-11
0	12-23
1	24-35
2	36-47
3	48-59
4	60-71
5	72-83
6	84-95
7	96-107
8	108-119
9	120-127

```
def calculate_octave_similarity(input_file, target_file, threshold=1):
    input_data_len = len(input_file)

    total_correct_octave_cnt = 0
    for index in range(len(input_file)):

        if index - threshold < 0:
            for i in range(0, threshold, 1):
                if target_file.at[index, 'new_octave'] == input_file.at[index+i, 'new_octave']:
                    total_correct_octave_cnt+=1
                break
        else:
            # threshold 만큼 앞을 탐색해서 있으면 점수 부여
            for i in range(-threshold, threshold, 1):
                if target_file.at[index, 'new_octave'] == input_file.at[index+i, 'new_octave']:
                    total_correct_octave_cnt+=1
                break

    return total_correct_octave_cnt / input_data_len * 100
```

옥타브 알고리즘 결과

수정 전 알고리즘

```
In [153]: octave_score = calculate_octave_similarity(input_file, target_file)
          print(f'octave 정확도 결과: {octave_score}%')
          octave 정확도 결과: 37.75%%
```



수정한 알고리즘

threshold = 1 일 때

```
print(f'octave 정확도 결과: {int(octave_score)}%')
octave 정확도 결과: 75%
```

threshold = 2 일 때

```
print(f'octave 정확도 결과: {int(octave_score)}%')
octave 정확도 결과: 87%
```

threshold = 3 일 때

```
print(f'octave 정확도 결과: {int(octave_score)}%')
octave 정확도 결과: 91%
```