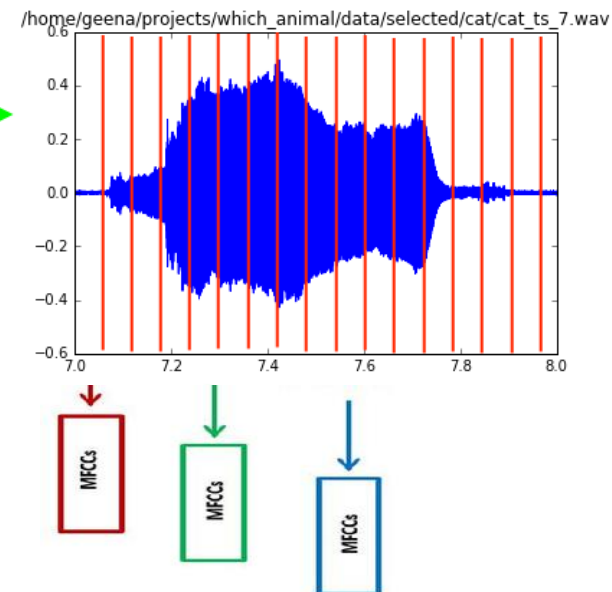
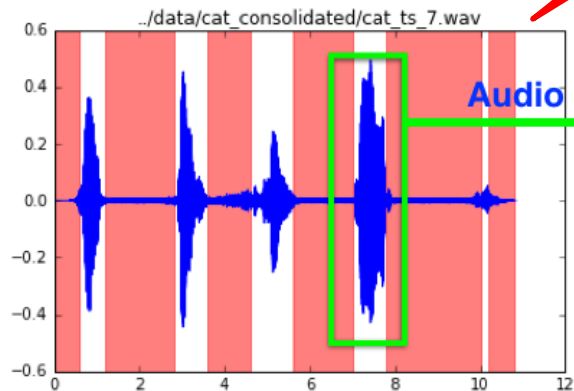




## k-fold cross validation

- 훈련 집합을 k개의 부분집합으로 나누어 사용. 한 개를 남겨두고 k-1개로 학습한 다음 남겨둔 것으로 성능 측정. k개의 성능을 평균하여 신뢰도 높임





# 혼동 행렬과 성능 분석

## ■ 혼동 행렬 confusion matrix

- 부류 별로 옳은 분류와 틀린 분류의 개수를 기록한 행렬

		참값(그라운드 트루스)					
		부류 1	부류 2	...	부류 $j$	...	부류 $c$
예측한 부류	부류 1	$n_{11}$	$n_{12}$		$n_{1j}$		$n_{1c}$
	부류 2	$n_{21}$	$n_{22}$		$n_{2j}$		$n_{2c}$
	...						
	부류 $i$	$n_{i1}$	$n_{i2}$		$n_{ij}$		$n_{ic}$
	...						
	부류 $c$	$n_{c1}$	$n_{c2}$		$n_{cj}$		$n_{cc}$

필기 숫자 인식 - 훈련 집합으로 학습하고 테스트 집합으로 성능 측정

```
[[76.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0. 78.  0.  0.  0.  0.  0.  0.  3.  0.]
 [ 0.  0. 66.  1.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0. 73.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0. 63.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0. 70.  0.  0.  0.  2.]
 [ 0.  0.  0.  0.  0.  0. 77.  0.  0.  0.]
 [ 0.  0.  0.  1.  0.  0.  0. 77.  0.  1.]
 [ 0.  0.  0.  0.  0.  0.  0.  0. 74.  0.]
 [ 0.  0.  0.  0.  0.  1.  0.  0.  0. 56.]]
```

```
16 # 혼동 행렬 구함
17 conf=np.zeros((10,10))
18 for i in range(len(res)):
19     conf[res[i]][y_test[i]]+=1
20 print(conf)
21
22 # 정확률 측정하고 출력
23 no_correct=0
24 for i in range(10):
25     no_correct+=conf[i][i]
26 accuracy=no_correct/len(res)
```