## < Tree ensembles - Random forest algorithm >

* Generating a tree sample

⇒ Given training set of size $m$

For $b = 1$ to $B$ : (= do B times)

- Use sampling with replacement to create a new training set of size $m$
- Train a decision tree on the new dataset

) repeat B times (보통 B=100)

(sampling with replacement를 통해 매번 새로운 training set으로 하나의 decision tree를 만듦
→ B번 반복 → B개의 decision tree를 얻을 수 있음)

⇓ 그런데 새롭게 얻은 training set으로 만든 decision tree들이
root node나 root 근처 하위 node도 같은 feature를 사용하는 경우가 있음

- Try to randomize the feature choice at each node that can cause set of trees you learn to become more different from each other ⇒ so that when you vote them, you end up with even more accurate prediction

* Randomizing the feature choice

- At each node, when choosing a feature to use to split,
if $n$ features are available, pick a random subset of $k < n$ features
and allow the algorithm to only choose from that subset of features.

⇒ 지금까지 Ear shape, face shape, whiskers 총 3개의 feature ($n=3$)로 decision tree를 만들었음
만약 100개의 feature가 있다면 (available), 100개 모두 사용하는 것이 아니라 random하게
특정 개수 ($k$)의 feature 만으로 여러 개의 decision tree를 만듦

= "Random forest algorithm" (typical choice of $k$ : $k = \sqrt{n}$)

⇒ classification : 여러 tree 결과 중 가장 많은 값
regression : 여러 tree 결과들의 평균값 (average)

Why this is more robust than a single decision tree?

- sampling with replacement procedure causes the algorithm to explore a lot of small changes in the data already and it is training different decision trees as averaging over all of those changes to the data

⇒ any little changes further to the training set makes it less likely to have a huge impact on the overall output of the overall random forest algorithm