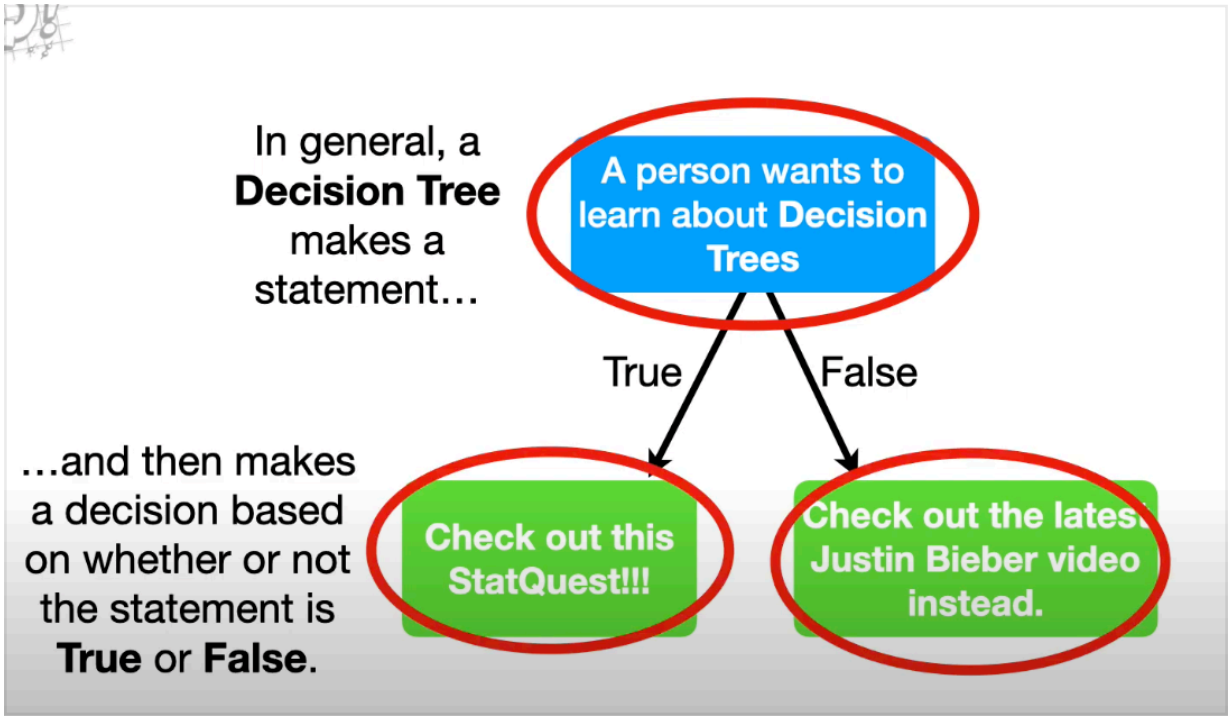
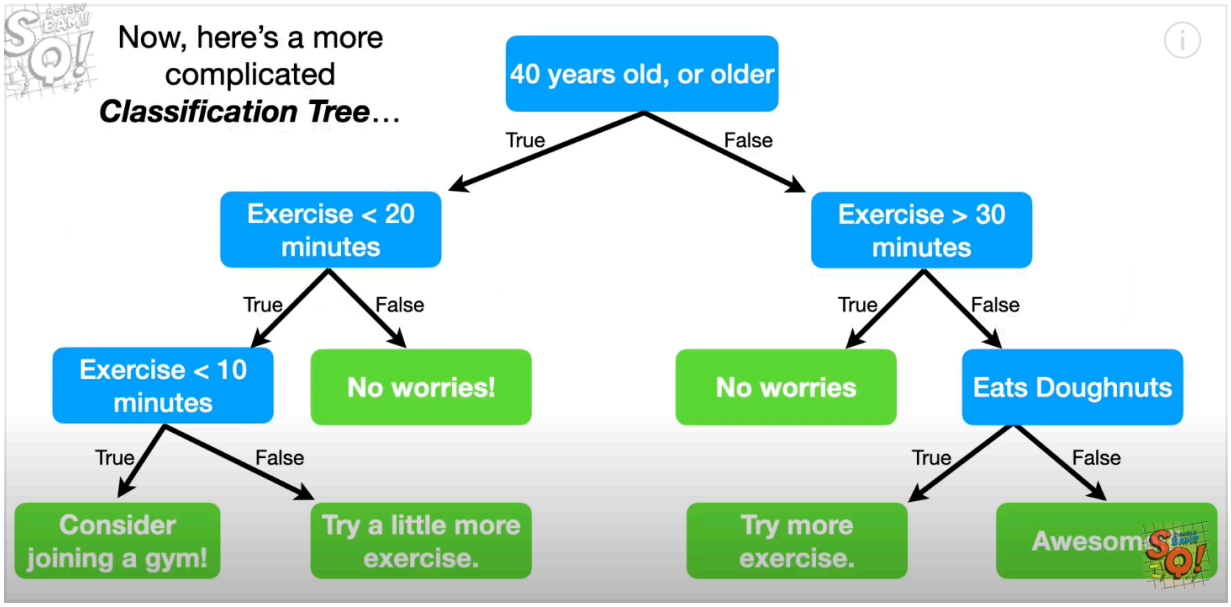
Random Forest CLF

Algoritmo de aprendizaje supervisado, por lo general genera una declaración y en base a eso, determina si es cierto o falso.
Ej:



Para clasificación se llama Árbol de clasificación; para regresión Árbol de regresión.

Ejemplo de un árbol de clasificación más complicado, combina declaraciones basadas en booleanos de operaciones numéricas (<, > & =). Va bajando cada vez más hasta que estoy en el límite y no puedo clasificando.
Por lo general las ramas verdaderas siempre van apuntando hacia la derecha de cada bifurcación; siendo lo contrario las falsas.



Para escoger el mejor split se tienen los conceptos de Entropia e Information Gain

$$Entropy:$$

$$H(S) = -\sum_i p_i(S)\, log_2\, p_i(S)$$

$$Information\ Gain:$$

$$IG(S,A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{S} H(S_v)$$

Intuitivamente, se debe entender como el camino a seguir el tratar de escoger un feature que mejor separa nuestra data, esto es tratar de maximizar nuestro "Information gain".

Random forest

Para mejorar el rendimiento podemos utilizar muchos árboles con una muestra aleatoria de features elegidas como la división.

- Una nueva muestra aleatoria de variables es escogida para cada arbol en cada split
- Para clasificación, m es comunmente escogido por ser la raiz cuadrada de p

¿cuál es el punto de escoger features aleatoreas?

Suponiendo que tuvieramos una feature muy fuerte en el dataset, si usásemos árboles ensamblados sin muestras de features aleatorias, muchos si no es que todos los árboles ocuparían esa feature para separar las hojas lo que ocasionaria un ensamble de árboles que están altamente correlacionados.
Promediar arboles correlacionados no reduce significativamente la varianza.
Dejando aleatoreamente features candidatas fuera a cada split, el Random Forest "de-correlaciona" los árboles, haciendo que el proceso de promediar pueda reducir la varianza del modelo resultante.