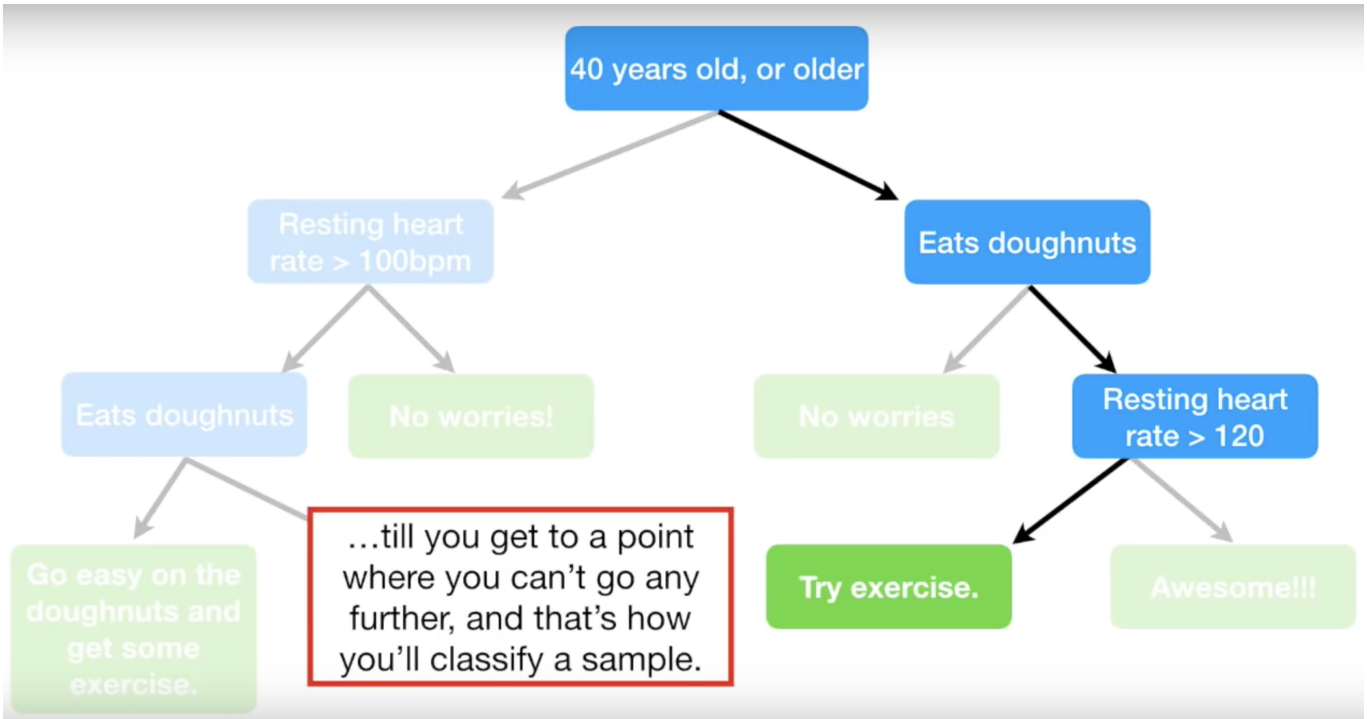
Decision Tree

Ein `Decision Tree` stellt eine Frage und klassifiziert dann die Person anhand der Antwort.

> Person hat einen hohen Puls - Person sollte zum Arzt
> Person hat keinen hohen Puls - Person muss nicht zum Arzt

Klassifizierung können Kategorien oder Zahlen sein!
Man kann auch beides kombinieren!



Namensgebung

Root-Node:
Die `root-node` befindet sich am obersten Ende des Baums. Von dort baut sich der Baum abwärts auf.

Node:
Sind die einzelnen `nodes` des Baumes.

Leaf-nodes:
Bilden den Abschluss eines einzelnen `strang`. Pfeile zeigen zu den jeweiligen `leaf-nodes` aber keine Pfeile gehen von dort weg.

Decision Tree aufbauen

Um zu entscheiden welcher der Faktoren die `root-node` betrachtet man wie gut jeder der einzelnen Faktoren den Outcome (das was man vorhersagen will) beschreibt. Da in den meisten Fällen keiner der Faktoren den Outcome zu 100% richtig beschreibt werden diese `impure` genannt. Diese Unreinheit kann zum Beispiel mit dem `Gini-Koeffizient` berechnet werden.

$$Für jedes Blatt - Gini Index = 1 - (WS für 'JA')^2 - (WS für 'NEIN')^2$$

Beispiel: Herzkrankhit vorhersagen

Root-Node: Brustschmerzen Leafe-Node1: (Herzkrank)

> Herzkrankheit Ja: 105
> Herzkrankheit Nein: 39

Leafe-Node2: (Nicht Herzkrank)

> Herzkrankheit Ja: 34
> Herzkrankheit Nein: 125

--> Gini-Index für Node1:

> $1 - (\frac{105}{105 + 39})^2$

--> Gini-Index für Node2:

> $1 - (\frac{125}{34 + 125})^2$

Um nun den Gesamten Gini-Index für `Brustschmerzen` als `root-node` zu bestimmen berechnet man die Anzahl aller Beobachtunghemn in dem jeweiligen Note gesehen auf die gesamt Anzahl von Node 1 un d Node 2:

$$ (\frac{144}{144 + 159}) * GINIIndex-Node1 + (\frac{159}{144 + 159}) * GINIIndex-Node2 $$

Das wird jetzt für alle Faktoremn gemacht (zum Beispiel noch von geblockten ASrterien). Der Faktor mit dem geringsten Gini-Imdex ist die `root-node`.
Der Baum baut sich nun nach und nach auf - sobald man die Daten nicht mehr trennen kann entsteht eine `leave-node`.

Random Forests

`random forest` basieren auf der `decision tree` Methode, kombinieren aber alle Vorteile davon mit einer Lösung für die Nachteile (schlechte Genauigkeit von Decsion Trees). Im grunde ist ein Random forest eine Vielzahl von gebootstrapten Decision Trees.

Wichtig dabei:
Für einen Random forest wird eine beliebige Anzahl von Parametern aus den Daten gewählt. **Doppelte Ziehung einer Zeile für einen decision tree ist ausdrücklich erlaubt!**

So werden ganz viele unterschiedliche Decision trees erstellt.

Um nun einenn Datensatz zu evaluieren wird dieser durch jeden dieser erstellten Bäume gejagt und alle Ergebnisse der einzelnen Bäume (ja - nein) aufsummiert. Die Mehrheit gewinnt und gibt das finale Label an.

Genauigkeit des Random Forest

Die Genauigkeit wird mit den `ausgelaseenen` Zeilen des Bootstrappings berechnet da man deren Ausgang kennt. Diese werden ebnfalls einfach durch den random forest gejagt und man kann somit dann eine Genauigkeit berechnen. In der Regel ist diese um einiges besser als bei einem einfach Entscheidungsbaum!