The problem with decision trees

Let's say we have a large table with lots and lots of columns. So, we create our Decision Tree. And we end up with answers like the following.

If a client is a male between 15 and 25 in the US, on Android, in school, likes tennis, pizza, but does not like long walks on the beach, then they're likely to download Pokemon Go.

This is not good. This almost looks like the tree just memorized the data. It's overfitting. *Decision Trees tend to overfit a lot.*

In the continuous case, this can also happen. The Decision Tree has many nodes which end up giving us a complicated boundary that pretty much borders every point with a small square. *This is also overfitting as it doesn't generalize well to the data.*

How do we solve this?

Pick some of the columns randomly from our data. Build a Decision Tree in those columns. Now, pick some other columns randomly and build a Decision Tree in those, and do it again. When we have a new data point, say this person over here, we just let all the trees make a prediction and pick the one that appears the most.

For example, these trees decided that this person will download Snapchat, WhatsApp, and WhatsApp. So, the ensemble of trees will recommend *WhatsApp*. Since we used a bunch of trees on randomly picked columns, this is called a **random forest**.

But, there are better ways to pick the columns than randomly.

Quiz Question

What is a consequence of Decision Trees that overfit?

a. They do not generalize well
b. They are computationally heavy