

Testing a maybe with machine learning.

24 July 2019 · ai, deep learning, investigation, machine learning

“I figured it was just a jumbo jet.”

My son and I shake our heads & then adopt blank stares as if a non-body-snatcher has been exposed in our midst.

“Twin engine,” I utter, as I glance skyward again.

“Single decker” My son adds as an explanation.

“It’s a plane”, she retorts, rolling her eyes.

My wife, (who is far smarter than myself) lacks my son and I’s ability to recognise aircraft. She has the typical persons ability to recognise aeroplanes. I grew up around airforce bases. I had a father who was an aircraft engineer. Years of exposure and explanations regarding aeroplanes, their mechanics and features.



We took this to the next level...

My son is an avid flight sim game player and has consumed many hours of relevant youtube material on the subject. He also had the luck/misfortune of me discussing the planes that frequent the skies, above us here, near London.

Given our combined experience & expertise, we probably have a reasonable ability to recognise the make & model of the planes we see in the sky. I would have given us, combined, an accuracy rate of say 90%.

A while back I worked on a Deep Learning classification model that could recognise types of aircraft from pictures. It was fairly crude and managed an 80% accuracy rate against a test set of aircraft images.

Is that good? That would depend on a lot of things:

- The purpose of the tool
- Alternative systems' performance
- The existing recognition system in place
- Risks the new system might expose a company to
- The cost
- The time/context when the device was used
- Ability to update the 'model' as new planes are released
- Etc

Those questions are probably similar to those you would be asking regarding any software you find yourself testing.

One subtlety that isn't always present in many software systems is the explicit accuracy. Often when a software system is performing a calculation or logical process, we assume 100% accuracy and test for it. That often makes sense, but sometimes those figures and calculations are inherently inaccurate from the user's point of view. They are just simplified models of real-world systems.

Contrary to much of the negative publicity Machine Learning receives, accuracy can often be measured directly with the model/tool. A machine learning approach could make the assumed inaccuracy of the tool, more explicit.

For example, we could get an overall idea of how accurate the model was (E.g.: 80% for our aircraft model) but also how sure it is (AKA the probability) of each answer it outputs (e.g.: its 95% sure its an Airbus A380, and 50% sure its a Boeing 747).

Furthermore, the data upon which it is trained can be defined and recorded for later review and analysis. This could be by programmers, testers, product owners, lawyers. prospective customers etc. That's not always easy to do if the existing system is a person.

As a tester, you might also locate or create your own data to more thoroughly test the system. Checking for edge cases and real-world situations that may have not already been modelled. E.g. What does the model classify a flock of Canada Geese as? Given realistic data, is the model biased towards giving Airbus planes a higher score? (we could test that...)

We can approach these systems as a form of mechanised heuristics. While they work slightly differently to our human-ware heuristics, they behave in a similar way. They are fallible, they are useful shortcuts that can really help in many situations.

For example, they could be replicated and deployed at will in a manner that existing people or systems can not. Will the product work better or more efficiently than the existing approach? (The answer, for example, it could be that it's less accurate than an existing person, but the unit cost is much lower - and so overall efficiency wins on a larger deployment)

While the roles being automated will still exist, there will be fewer people doing them. E.g.: Why have ten aircraft spotters when 2 who are skilled in both identifying the UFOs and updating the machine learning models might fit the business needs more efficiently?

As we continue to mechanise more business functions using machine learning, Software Testers will need to start thinking a little more in terms of comparing accuracy rather than a brittle approach of binary pass/fail correctness. Given our industry's struggles with the Pass/Fail mentality, I suspect this will be one of our greatest challenges in the coming years.