ttc

Machine Learning for Test Automation

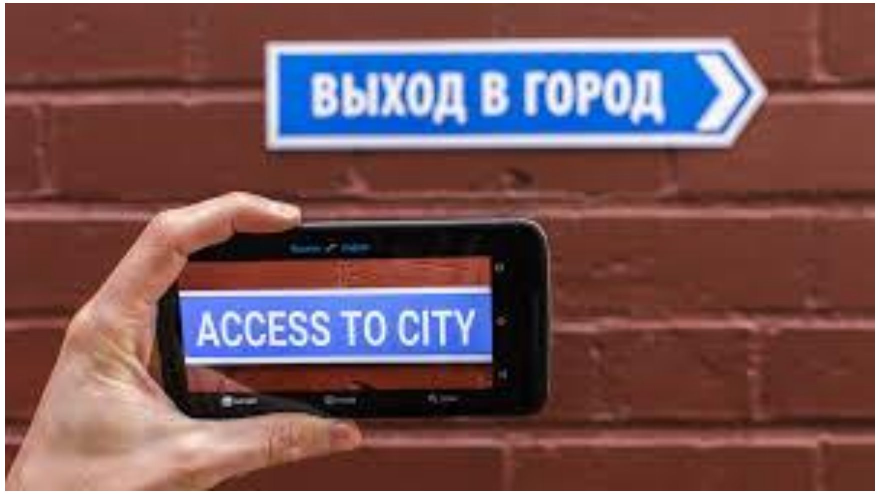
Sam Fernando - May, 2017

Agenda

- What is machine learning?
- Application to test automation
- Recommendations

What is Machine Learning?

What is artificial intelligence?



Is this a smart phone?
(does it have intelligence)

What is human intelligence?

Machine Learning 101

AI and Machine learning

- An intelligent agent is “any device that perceives its environment and takes actions that maximize its chance of success at some goal”
- Machine Learning is the training of a model from data that generalizes a decision against a performance measure
- ‘Learning’: Getting better at a task over time (i.e. takes into account new data points)

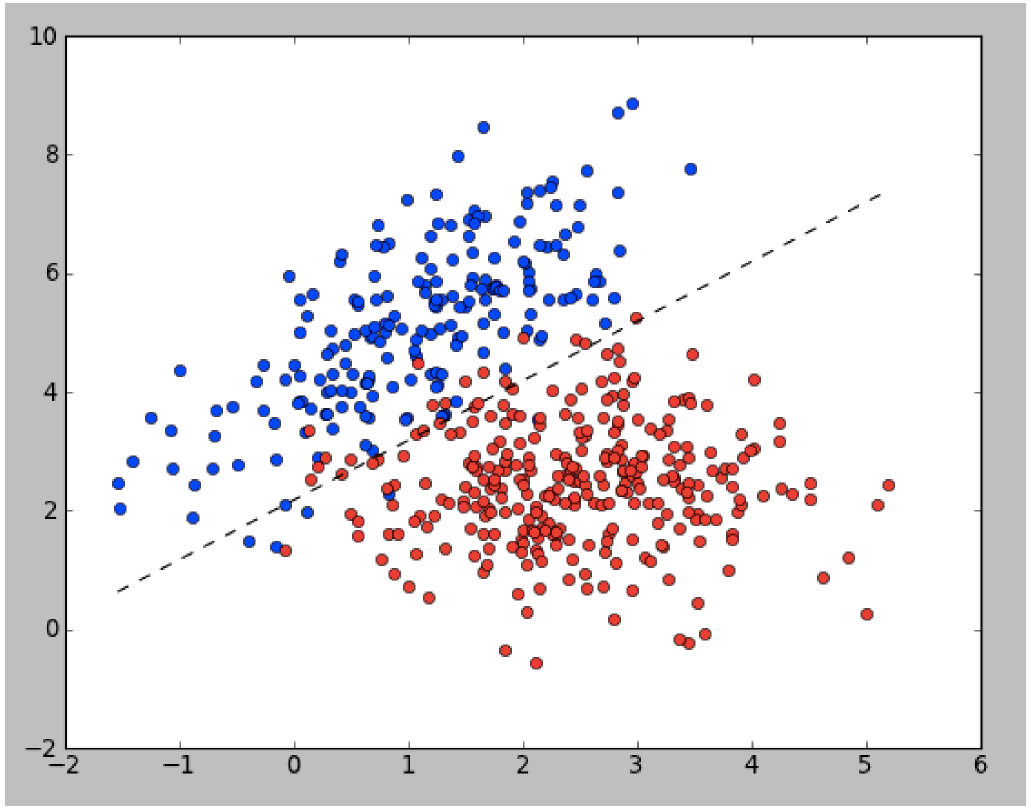
Machine Learning 101

Supervised learning

- Learn from test data
- Make predictions
- Data is labelled

Examples

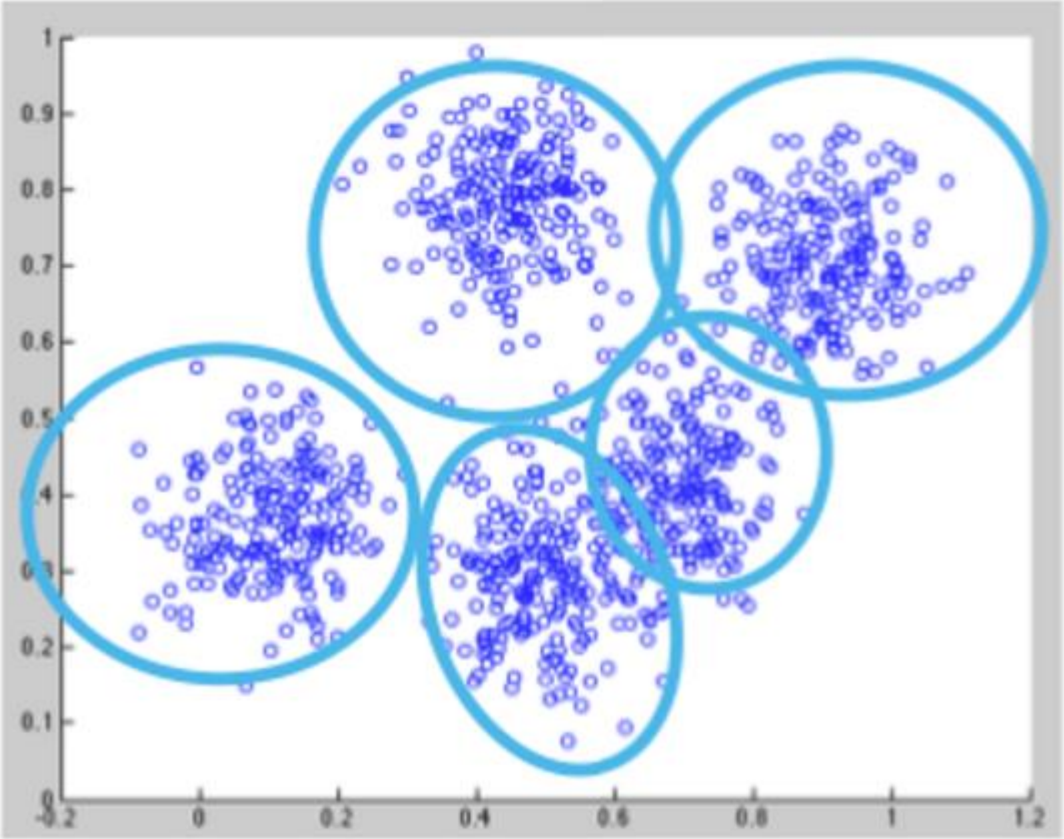
- Email spam filter
- Image recognition



Machine Learning 101

Unsupervised learning

- Discover structure within the data
- Data is unlabelled



Examples

- Google news
- Social network analysis

What we have been working on

How could machine learning help with testing?

- Natural language processing: turn requirements/stories into tests?
- Create new test data based on defects found?
- Automation maintenance: Recognise controls even when they change

What’s the problem?

- When a control – say an edit box on a web-page – changes, we, as humans, can see what control our test script it meant to steer. But our scripts cannot.
- This leads to false negatives:
  - Lost execution time
  - Increase analysis time
  - Increased maintenance time

What’s the problem?

1. Test automation uses properties of controls to find and steer them

2. Controls on a web-page change as they are developed

→ Therefore test automation scripts won’t be able to find the controls

Machine learning solution:

- How can we get our automation script to learn what the right control is even when it has changed in some respects?

Machine learning process

- The web-archive stores the html of each iteration of web-sites
- Use selenium to scrape the html elements, and their properties, for historic version of the same web-site

Example:

https://www.kiwibank.co.nz/ → CSV file

Step 1: Gather Data

- The web-archive stores the html of each iteration of web-sites
- Use selenium to scrape the html elements, and their properties, for historic version of the same web-site

Example:

https://www.kiwibank.co.nz/ → CSV file

Step 1: Gather Data

Simplify the problem to just input elements

Demo

Step 2: Choose an algorithm

- We need to pick a model which represents the patterns in the data
- The algorithm will then optimise the model’s parameters to suit our data
- Would normally pick multiple models, and find the best one

We will use supervised learning: linear regression

- Label which controls are matches to other controls
- Look at each property independently

Step 3: Pre-process the Data

- Use supervised learning:

→ Label which controls – across different web site iterations - are in fact the same as each other

- Machine learning algorithms only deal with numbers not strings

→ Compare how similar each property of a control is to the same property of the other controls, assigning a number between 0 and 1

Step 4: Train the Classifier

- The algorithm, Linear Regression, learns how important each property is in determining whether or not a control is a match

Demo

Step 6: Keep learning

- When the classifier is used, feedback if it picked the correct control
- That becomes a new piece of data which can be fed into the algorithm, which refines the parameters of the classifier’s model

Step 5: Make Predictions

- The classifier can now be ran against a web-page of controls, comparing each control to the one our test script is looking for
- It returns how likely it is that each control is a match
- The tool, here Tosca, will execute the test step on that control

Demo

Recommendations and Limitations

- Lack of generalisation
  - How one control is changed, by one developer, for one web-site, could be totally different from another
  - Even with more data, better pre-processing (i.e. ‘submit’ ~ ‘login’), and non-linear relationships between inputs, it is unlikely that a general pattern exists.
- Effort would be better spent on a pure rules-based best match solution: filtering, cases, weighted similarity
- Using automation best practices would resolve most solvable robustness issues: repeatability, timing, recovery

Questions and Discussion