

# Fantastic Problems

And Where to Find Them

Daryl Weir\_  
Senior Data Scientist\_

# We help our customers succeed in digital business\_

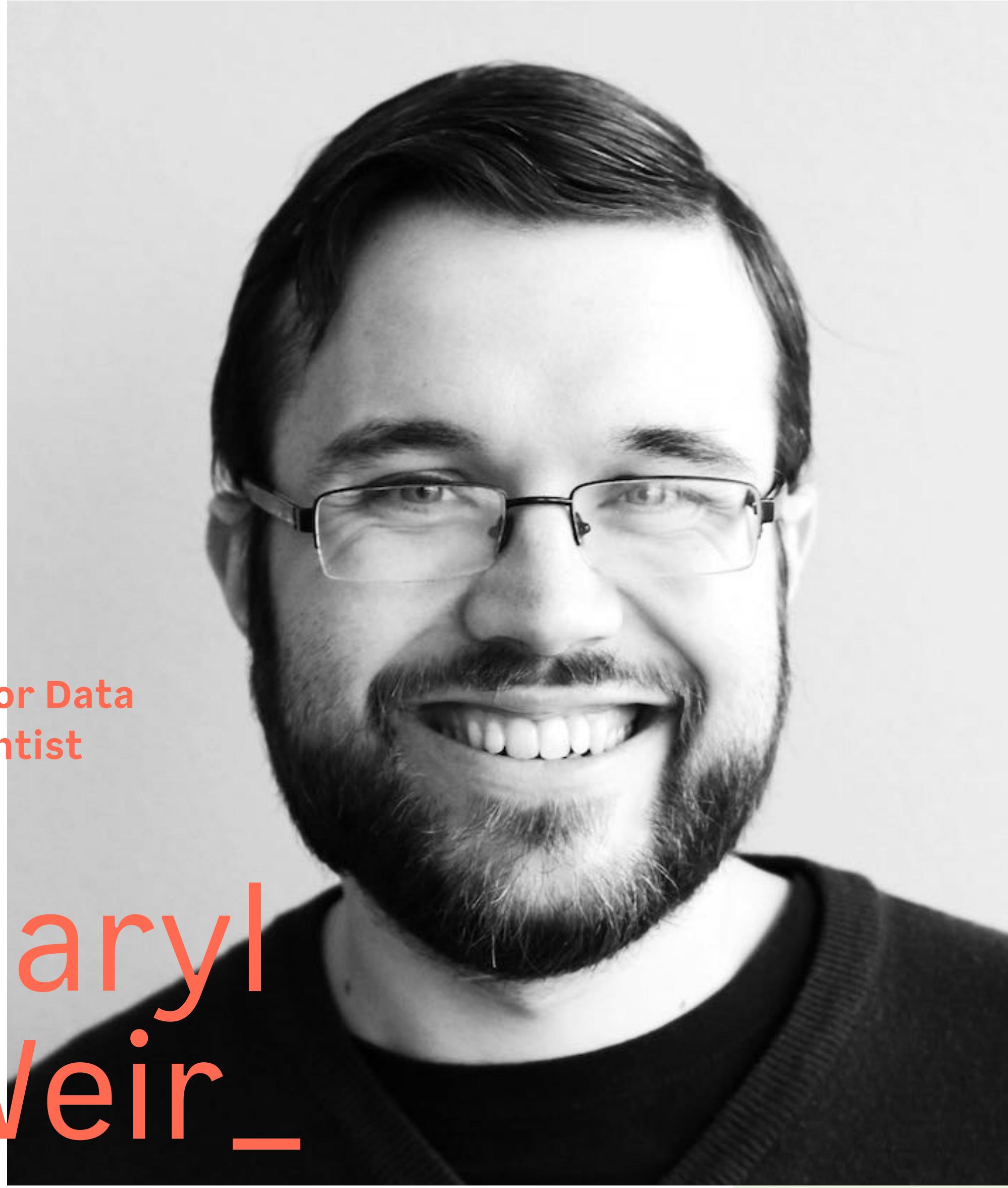
Founded in 2000

300+ employees from 22 countries

8th year in a row profitable growth

YOY growth 30%





**Senior Data  
Scientist**

**Daryl  
Weir\_**

**PhD Computer Science** 2010-2014  
University of Glasgow

**Research Intern** 2014  
Microsoft Finland

**Postdoc Researcher** 2014-2016  
Aalto University

**Data Scientist** 2016-Now  
Futurice

daryl.weir@futurice.com  
@darylweir

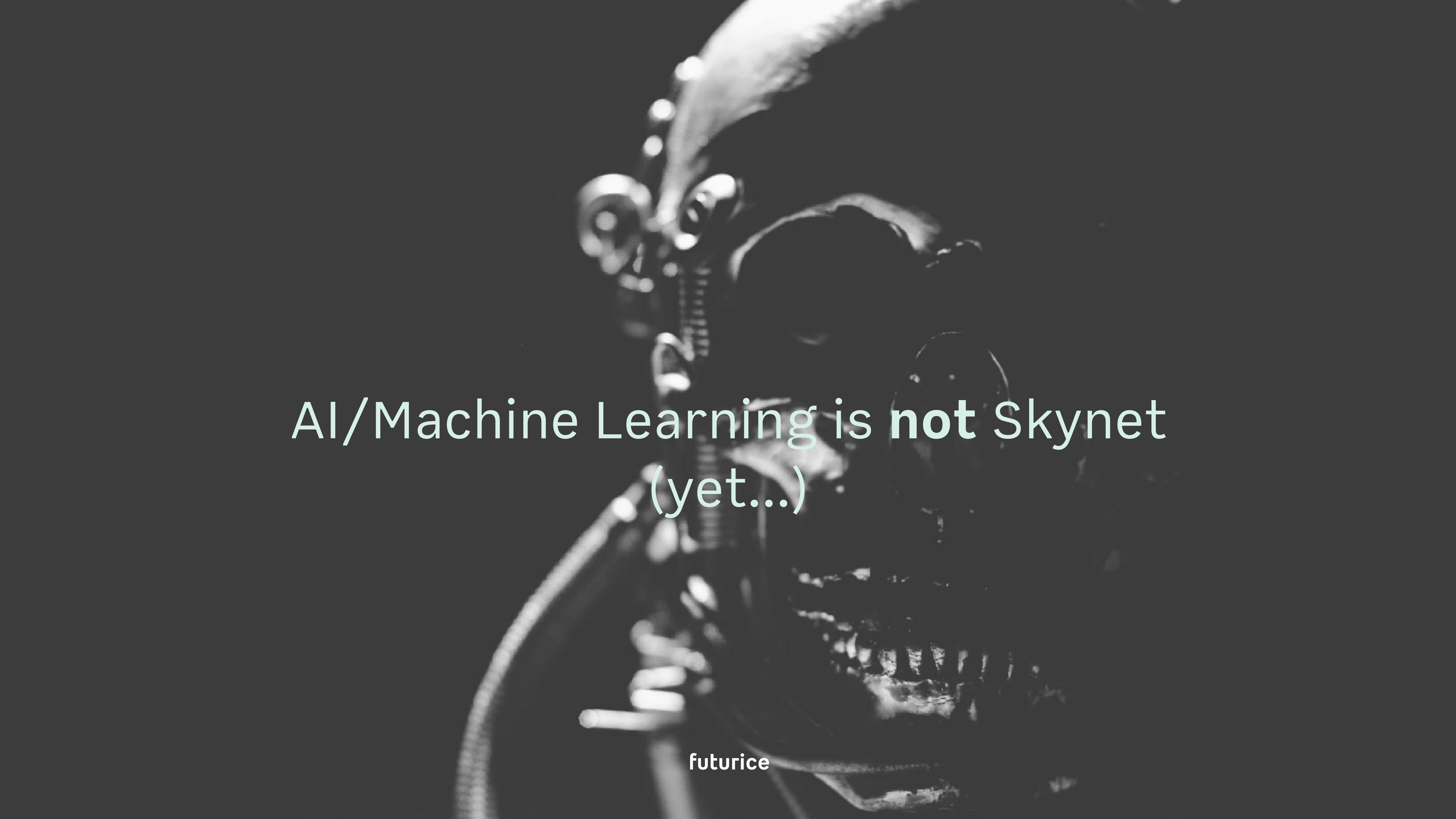
# Machine Learning Hype\_

AI, big data, machine learning... These have been buzz words for years now

This talk tries to answer two key questions:

- **What can machine learning actually do?**
- **Should I use machine learning to solve my problem?**





AI/Machine Learning is **not** Skynet  
(yet...)

# So what is it?\_

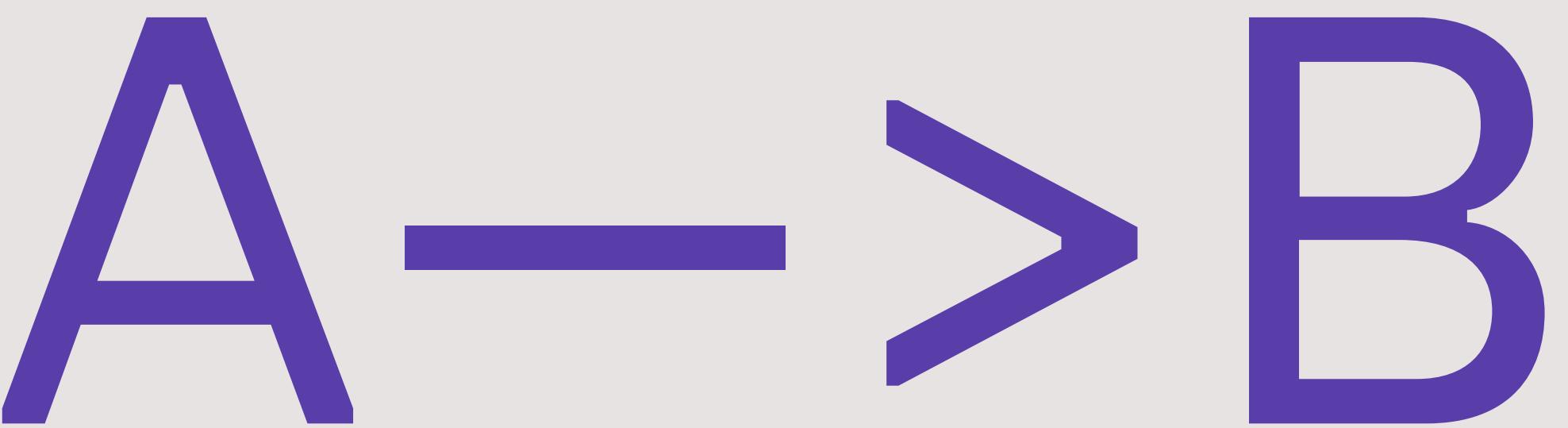
Machine learning is really good at  
**answering narrow questions**

It does this by **learning from examples**

The most common method is **supervised learning**, where problems look like:

Given some **input A**, predict an **output B**

By choosing the right A and B, you can do  
some amazing stuff



# For example\_

Baby's first self driving car

**Narrow question:** given the road looks like this, how should I turn the steering wheel to not crash?

**Input:** camera image of the road

**Output:** steering wheel angle

(CMU did this in 1995 - the car drove 3000 miles across the US)

But what kind of problem should we  
look for?\_

**“If a typical person can do a  
mental task with less than one  
second of thought, we can  
probably automate it using AI”**

- Andrew Ng

# A (somewhat trite) counterexample\_

Are these numbers **odd or even?**

You can (hopefully) judge this in under a second

But many popular machine learning algorithms are **terrible** at learning this (from the numbers alone)

Not all problems that **can be** tackled with ML **should be**

18

322

131

1061

94

27

# My rule of thumb\_

You might have a machine learning  
problem if:

1 - writing down a set of rules is hard

**BUT**

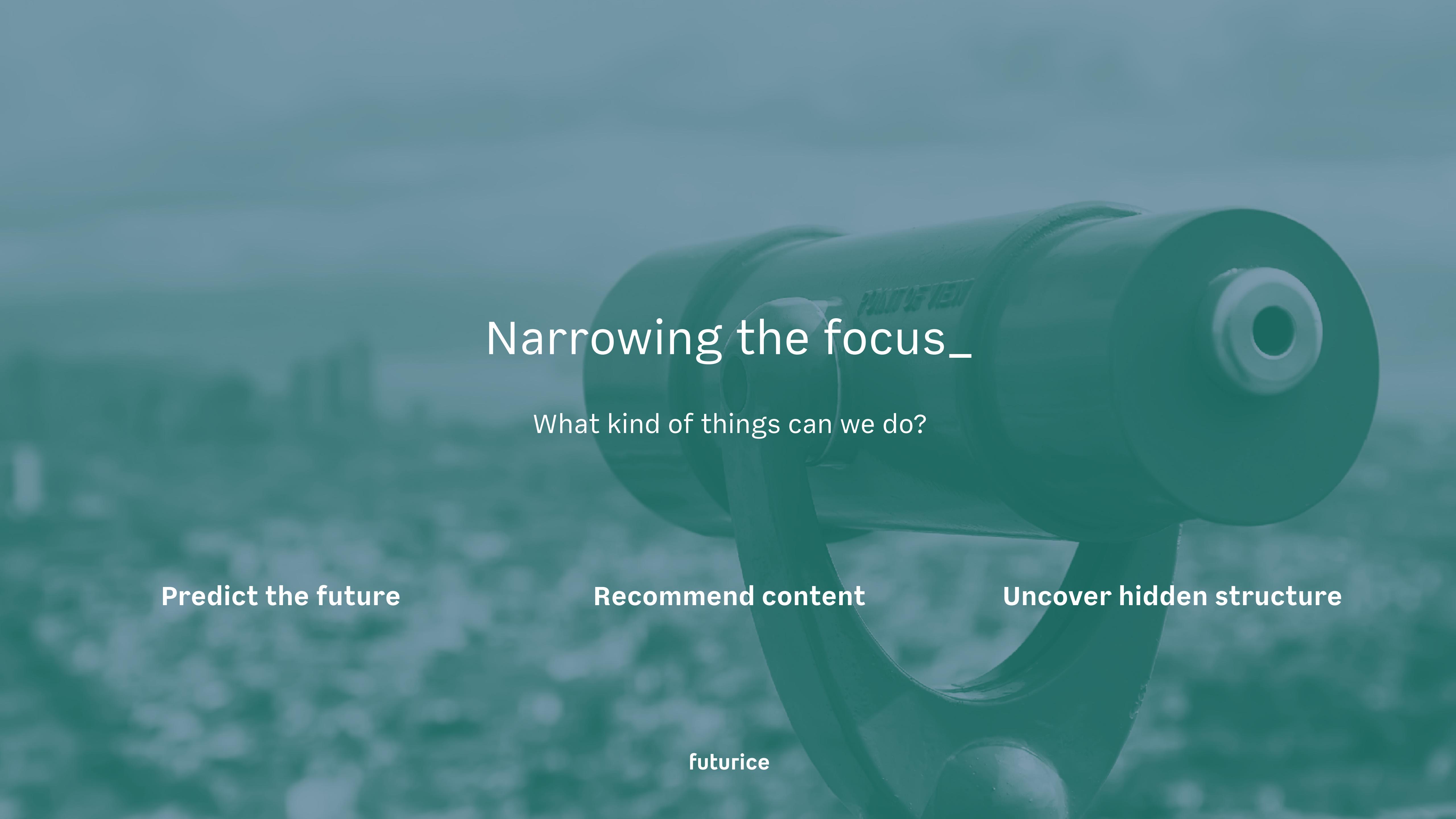
2 - gathering examples is easy

# Is there a cat in this image?\_

Writing rules for the presence or absence  
of a cat is really, really hard

**BUT**

Finding cat pictures is really, really easy



## Narrowing the focus\_

What kind of things can we do?

Predict the future

Recommend content

Uncover hidden structure



# Predicting the Future\_

futurice

# Scenario\_

We have lots of historical examples of system state, each one labelled by its outcome

Given a new example of the state, we try to predict the outcome

Such problems fall into two main types: **regression** and **classification**

**Regression problems** are those where the outcome is a number, for example:

- how many potatoes will my supermarket sell next week?
- what's the optimal price to sell this stock?
- what speed should I drive given the weather & traffic?

**Classification problems** are those where the outcome is one of a fixed set, for example:

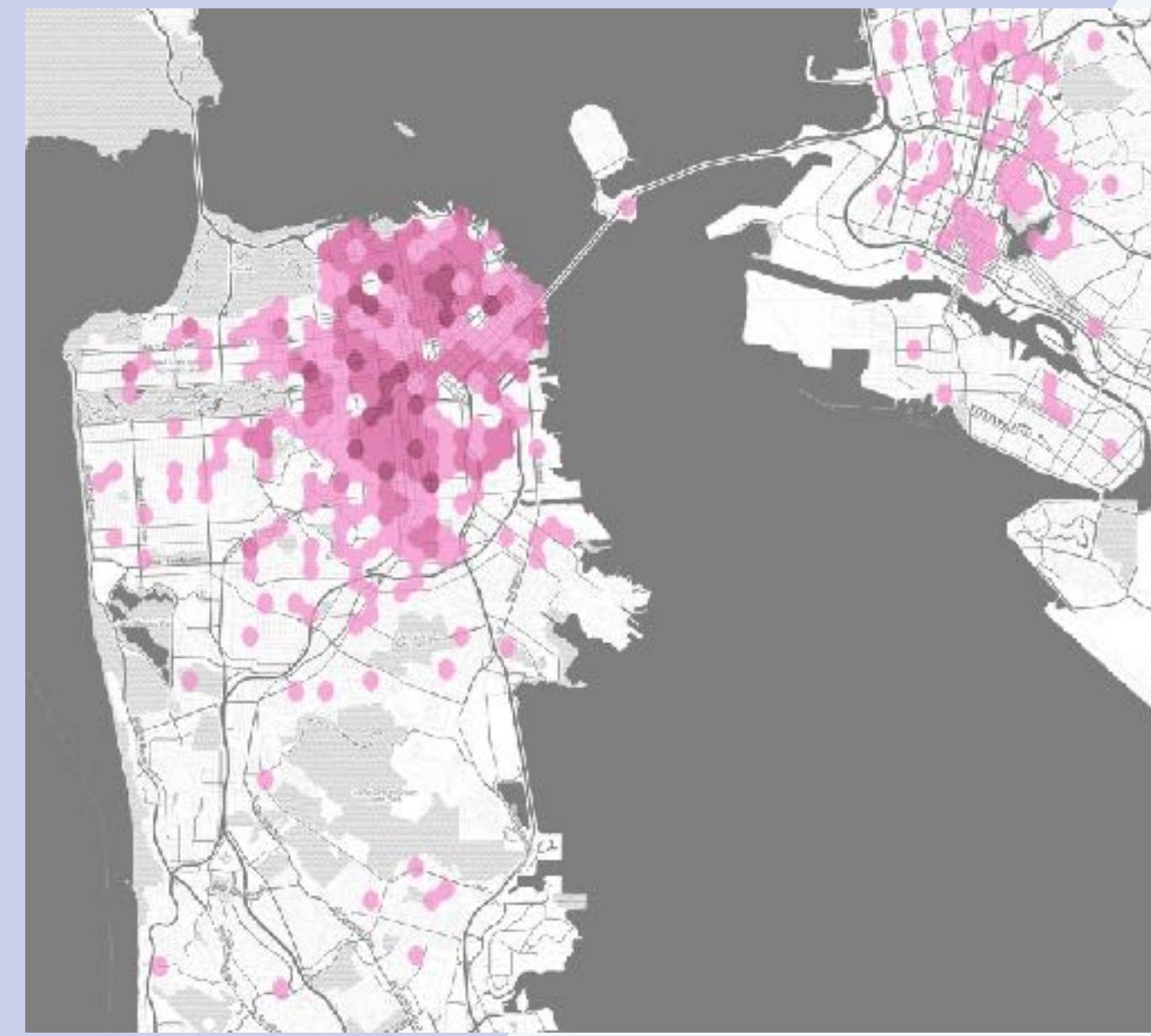
- whose face is in this image?
- is this credit card transaction fraudulent?
- do these scan results indicate cancer?

# Case: AirBnB\_

AirBnB is a great example of a company oriented around data

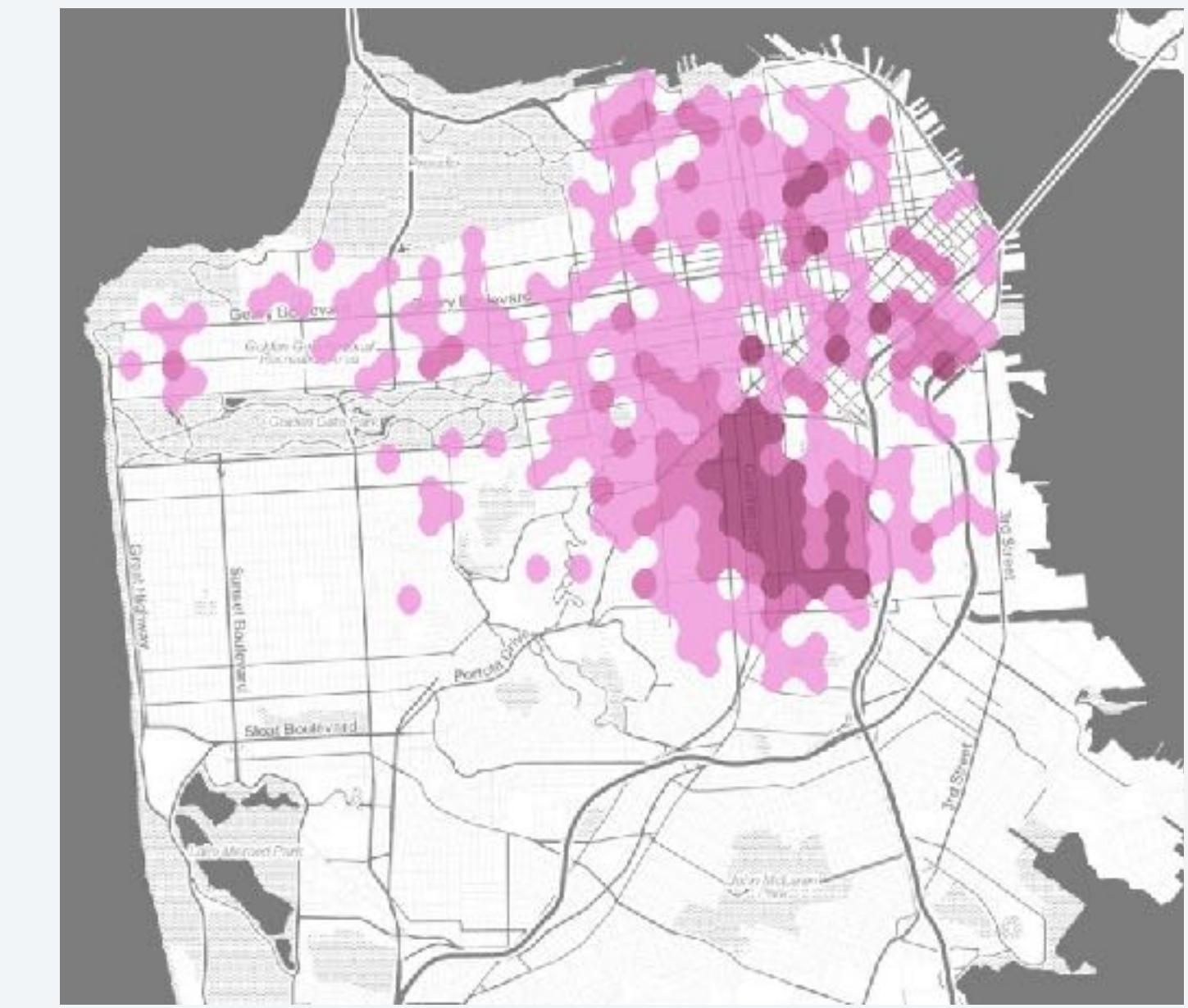
They blog regularly about data science and open source some of their tools

One problem they've talked about: given the user's search term, **should we show a given listing in the results?**



Basic distance based search for San Francisco shows many rentals in Oakland

Users were likely to find these results irrelevant



A classifier predicts which apartments will be booked after a specific search

This sped up the booking process and significantly increased conversion rate

# Case: Railways\_

Two buzz words for the price of one: machine learning + IoT!

Modern trains and rails are highly instrumented with sensors

From this data, we can predict whether a part (e.g. a wheel or door) is **about to fail**



Lets the operator move from scheduled to on-demand maintenance/replacement

VR Group estimates this cuts amount of wheel maintenance by around a third

**Bonus inception:** sometimes the sensor themselves start giving incorrect readings

Another classifier can detect these faults before they suggest phantom breakdowns



Recommending Content\_

futurice

# Scenario\_

Many businesses depend on content of some sort, e.g.:

- articles and videos
- products in a store
- ads

It is often valuable to personalise the content shown to your users

There are two main methods to **predict content users will like**

**Content based recommenders** suggest, unsurprisingly, based on the actual content

- these articles have similar topics to your previous reads
- more videos from channels you watched

**Collaborative filters** suggest things other users have liked

The preferences of others are weighted by how similar they are to you

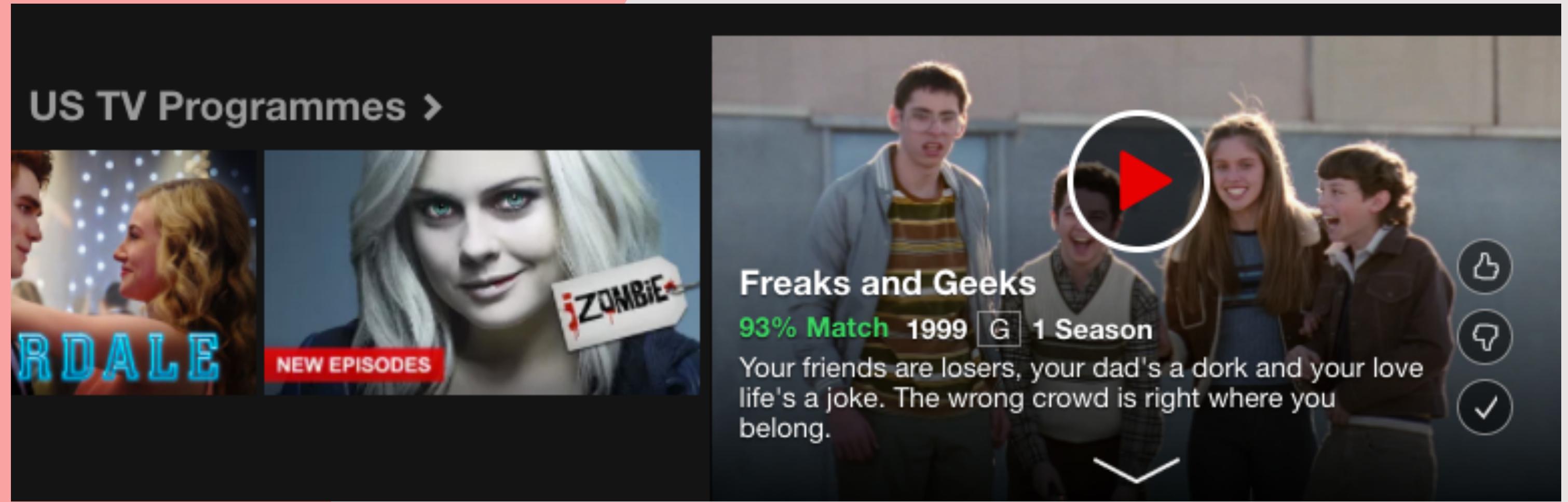
- “Customers also bought...”
- Spotify Discover Weekly

# Case: Netflix\_

Netflix is one of the most well known users of recommender systems

Like AirBnB, they blog regularly and have some open source tools available

Good recommendations increase user satisfaction, encourage continued use, and drive users to the ‘long tail’

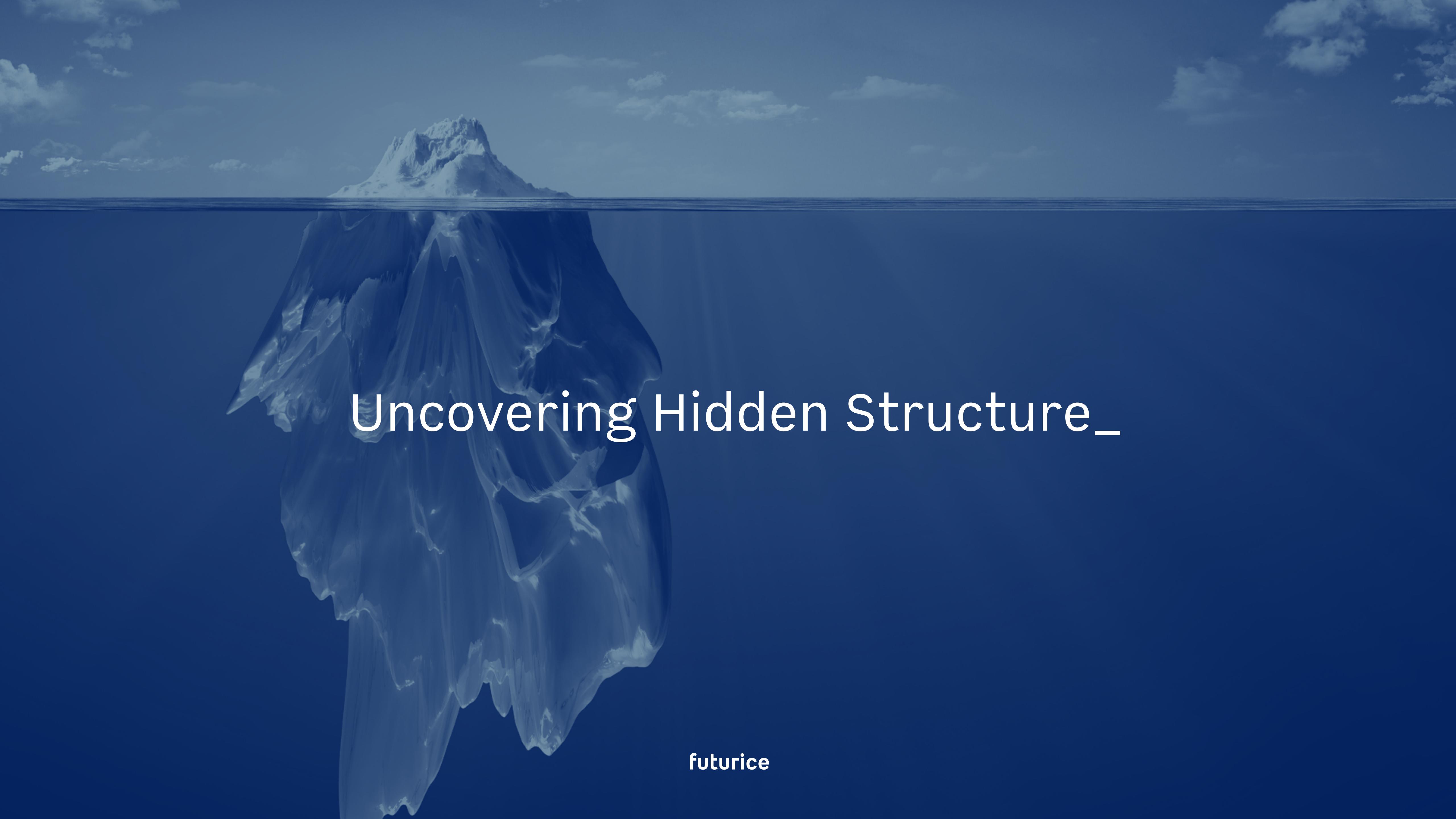


Recently changed from star based ratings to a thumbs up/down system

Better UX, avoids differences in users' rating systems

Famously offered a \$1M prize to anyone who could improve their recommender by 10%

Winning solution was never used!

The background image shows a massive iceberg floating in a dark blue sea under a light blue sky with scattered white clouds. The iceberg is mostly submerged, with a small portion visible above the waterline, symbolizing hidden complexity beneath apparent simplicity.

# Uncovering Hidden Structure\_

futurice

# Scenario\_

What if we don't have labels for our historical states?

Another major ML application area is automatically detecting recurring patterns in data

This is called **unsupervised learning** - it's about learning what's interesting about some given data

This lets us ask different sorts of questions, like:

- how do players of my game **group together** based on play behaviour?
- is this spending behaviour **unusual** given someone's past purchases?
- what are the **common topics** in this collection of documents?

# Case: Google News\_

Google has been very vocal about its transformation into “a machine learning company”

As of 2016, ~10% of their engineers have ML experience and internal training is growing

They have stated a goal of systematically applying machine learning in all their business areas

The screenshot shows the Google News homepage. On the left, there's a sidebar titled "World" with a list of topics: Mike Pence, France, Australia, Venezuela, Syria, USS Carl Vinson, Iran, Donald Trump, China, and Nuclear weapons testing. To the right of the sidebar is a main content area. At the top of this area is a story from the Daily Mail about Tim Farron being challenged by a fish finger in an election. The story includes a photo of Tim Farron, a photo of a fish finger, and links to other news sources like ITV News and NW Evening. Below the story is a note stating, "The selection and placement of stories on this page were determined automatically by a computer programme."

Google News automatically groups versions of the same story from different outlets

Stories are also grouped to form news categories

The rankings of different topics and outlets can be tailored by the user explicitly

The service can also learn user preferences from usage data

A black and white photograph showing a close-up of a person's hands typing on a laptop keyboard. The person is wearing a light-colored long-sleeved shirt. The laptop is open, and the keyboard is visible. The background is blurred.

Trying it out\_

# Starting out with ML\_



Excellent tooling - try the  
Anaconda distribution

Jupyter notebooks allow mixed  
code and Markdown

Can be slow



Designed with data analysis in  
mind

Amazing library and  
community support

Not general purpose



Only for masochists

# Resources\_

“I want tutorials and interesting blog posts pitched at beginners (and I don’t mind Python)”

[github.com/hangtwenty/dive-into-machine-learning](https://github.com/hangtwenty/dive-into-machine-learning)

“Python sucks, show me resources for machine learning in <my language of choice>”

[github.com/josephmisiti/awesome-machine-learning](https://github.com/josephmisiti/awesome-machine-learning)

“Coding sucks, I just want to read a book”

‘The Master Algorithm’ by Pedro Domingos

# If you remember 3 things\_

- 1) Machine learning answers narrow questions
- 2) Look for complex rules with plentiful examples
- 3) Don't use Matlab

Thanks for listening!  
[daryl.weir@futurice.com](mailto:daryl.weir@futurice.com)  
[@darylweir](https://twitter.com/darylweir)