INTELLIGENCE AUGMENTATION: A DESIGNER'S HANDBOOK

2.0

iadesignkit.com

Learning to Dance with Machines

The future of work is a real-time dance of human and machine intelligence. Service Designers need to be conversant in the language and concepts of machine intelligence to create complete packages that enhance life.

As digital technology makes deeper inroads into our lives, designers face heavier social responsibilities. Technically and ethically, future services and products will have to perform on an unprecedented level of transparency, reliability and intelligence.

The presence of machine intelligence in the objects around us is resulting in a proliferation of services and products that will wreak physical and financial havoc when they fail or misbehave. Consumer grade will go from being cheap to being both financially and ethically unaffordable.

Artificial Intelligence (AI) uses machine learning to replace human behaviour (e.g. self-driving cars). Intelligence Augmentation (IA) uses machine learning to create smart assistants that extend and enhance human capabilities (e.g. Google search). Both of these techniques will increasingly impact the services and products we all use and therefore the way they are designed. This booklet, and the associated workshop, will help designers to become conversant with the opportunities and threats machine learning services present.

Machine Learning and Design

Machine learning is the technology at the heart of the new breed of smart services. It is a collection of techniques designed to **learn rules from examples**. In regular software, programmers write the rules themselves as step-by-step instructions for the computer. Machine learning lets the computer figure out the rules for itself by looking at examples of the correct behaviour.

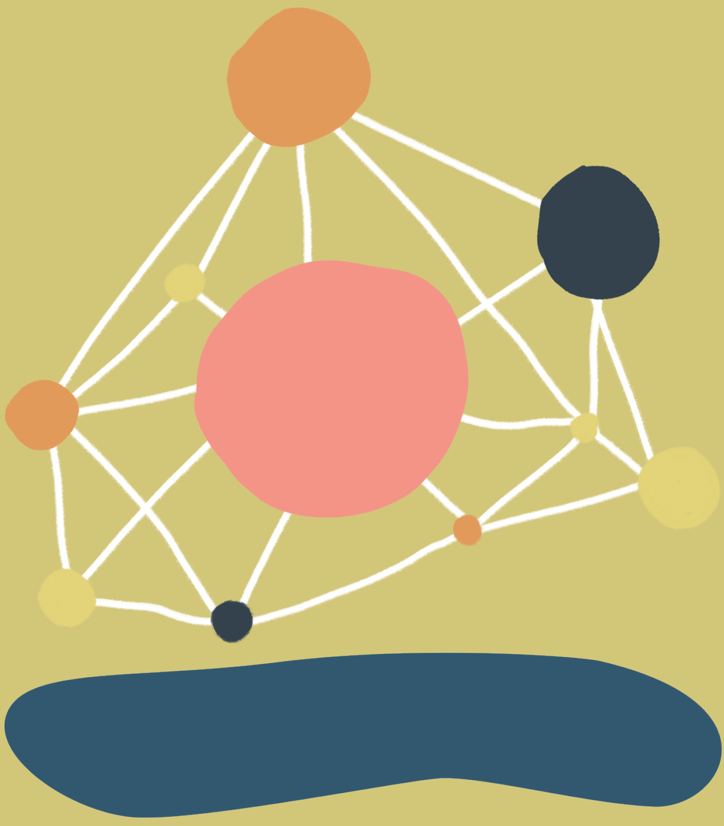
Smart machines are good at just one thing. They don't learn to perform many different tasks as humans do. If I show my self-driving car a chess board, it won't learn to play. Instead, for each different task, the machine has to be told exactly what to learn, and we need to understand that it will sometimes make mistakes. That is **where design comes in**.

Machine learning is only useful when applied to **problems worth solving**. Finding those problems which are also appropriate for machine learning still requires classic design processes - user research, concepting, and a few new techniques. Think of machine learning as a **toolbox** that gives you the designer new ways to address human needs.

Not every problem can, or should, be solved with machine learning. Here are a few rules to identify problems that might be good candidates:

1. A human expert could perform the task in a few seconds
2. It's difficult or impossible to write down the rules
3. It's easy to get examples of the desired behaviour
4. Knowing the rules would enable actions to be taken

For example, consider self-driving cars. Drivers make a constant stream of decisions based on the road and traffic, each of which takes a matter of seconds (see 1, above). However, it's impossible to write down step-by-step rules for all the situations that come up while driving (2). Despite this, it's easy to have a human drive around, and record how they turn the wheel, operate the pedals, and so on. Combining this with video of the road gives a huge amount of example data (3). Machine learning can study this to create a set of rules for the car to drive itself (4).



Machine Learning Use Cases

Many companies are already successfully employing machine learning in their applications. Popular use cases can be loosely grouped into four categories:

Predict

Predict something about the future. The prediction can be:

- A number (e.g. sales volume, stock price)
- A yes/no answer (e.g. will this hard drive fail tomorrow?)
- One from a set of 3+ options (e.g. is this patient high, medium or low risk)

Personalize

Tailor the behaviour of a system to specific users or groups of users. This is usually:

- Recommending content (e.g. written articles, physical products, ads)
- Targeting communication (e.g. targeted email campaigns, individually tailored UI)

Recognise

Identify information from input signals, such as:

- Images (e.g. face/object recognition)
- Sound (e.g. speech recognition, song identification)
- Text (e.g. chatbots, document summarisation, text translation)

Uncover Structure

Identify interesting patterns and information in data. Examples include:

- Group discovery (e.g. user segmentation, product similarity map)
- Discover unusual behaviour (e.g. fraud, sound from failing machinery)

Key Issues

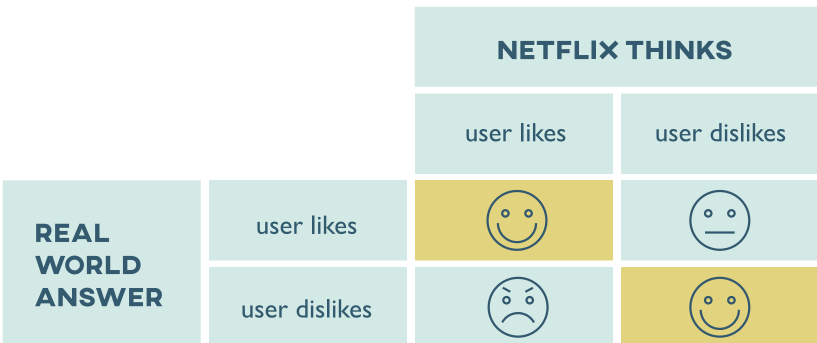
Adding machine learning to a service introduces many new considerations to the design process. Some of the most important are discussed below.

Designing for Failure

No machine learning technique is 100% accurate except on trivial problems. The accuracy will depend on the combination of technique and the collection of examples it learns from (the **training data**). This means there is a fundamental uncertainty about the state of the system, and you don't have perfect control over every user's path through the application.

This is a key design challenge when working with machine learning. Errors can and will happen, and you have to plan for this. Moreover, there are different kinds of error, and depending on the application these will have different impacts on the customer.

One useful tool for reasoning about this is the confusion matrix. It lists the possible decisions the machine might make, and compares those to the different cases that might happen in reality. For example, consider what happens when Netflix recommends TV shows and movies- they have to predict whether or not the user will like each item.



When the real user preference and Netflix's prediction line up, everything is fine. The user gets recommendations they like, and doesn't see recommendations for things they dislike. There are two different error cases:

- The user would have liked an item, but Netflix doesn't recommend it.
- Netflix recommends an item but the user doesn't like it.

The first case is effectively invisible to the user; they don't know the content was there in the first place. The second case has a higher cost, since it's annoying to get recommendations for things you don't like.

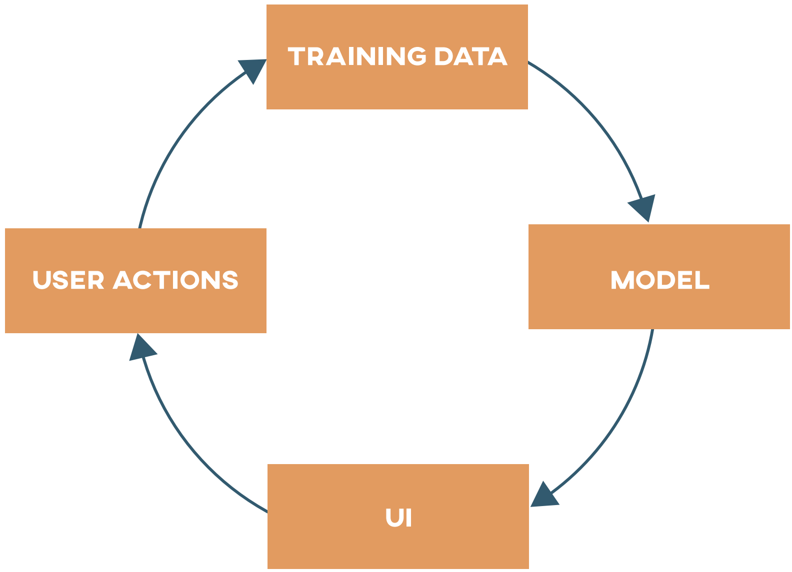
This information can be passed from the designer to the data scientists making the algorithm - most machine learning techniques can be made to favour one kind of error over the other. This lets the overall experience be made to minimise the cost to the user.

Filling the matrix for your specific ML application gives valuable estimates of the cost to the user or service of different errors, and also lets you plan interaction flows to let the user continue when an error occurs.

Designing for Learning

Feedback loops are nothing new in design. Building a product, testing it, and iterating based on what you learn. Working with machine learning brings an interesting twist. In addition to you as a designer learning about the system at each iteration, the system itself can also learn from its users and become better over time.

This is especially relevant in Intelligence Augmentation services, where the machine learning acts as part of a user-facing service. Every time the system makes a prediction, it is shown in the UI, and based on that the user will take an action. If we can **measure** that action, we can use it as additional training data for the model in the future. This is the **interaction loop** of the service.



Designing an effective interaction loop is important. One obvious design is to add explicit feedback - for example, a thumbs up or down button to indicate when a recommendation is good or bad. However, you can't rely on people to use such features. It's more powerful to leverage implicit feedback, where you interpret the meaning of regular interactions. For example, if you recommend a video and someone clicks and watches it, that indicates the recommendation was good. If they never watch it or start watching but give up quickly, it was probably a bad recommendation for this situation. By recording such actions, data can be built up and used to improve the recommendations over time.


UNEXPECTED BUG
Network Crash - System loses access to the Internet for an extended period.

Designing for the worst

A machine learning algorithm is only as good as its training data. If the data contain inherent bias, the algorithm will learn to repeat and reinforce that bias. One notorious example was the first rollout of the object recognition feature in Google Photos. The service incorrectly labeled people with dark skin, causing controversy online. This could have been prevented if the training data had contained a more diverse set of faces.

As a designer of IA and AI, one of your responsibilities is to consider how bias could appear in your service, and make sure that the training data is diverse enough to prevent that. If possible, design sanity checks that can catch potential bias before the service is released to external users.

If you design a machine learning system with an interaction loop, one possible consequence is that the behaviour of the system will be reinforced every time around the loop. This will tend to make the predictions for a given user or user group more and more similar as time goes on. This creates a **filter bubble**, where people see more content that matches their existing tastes.

In some applications, filter bubbles are fine - getting better music or tv show recommendations doesn't really hurt us. In others, bubbles can be a severe problem. This effect in social media has created echo chambers that perpetuate stereotypes and increase political division. It's important to consider what the consequences of this reinforcement will be in your own application, and decide if filter bubbles are a problem. If so, talk to your data scientists. It's possible to add some amount of randomness or intentional error to the machine learning, which can slow or prevent the formation of these bubbles.

Dictionary

Algorithm / Learner
A single mathematical or statistical technique from machine learning. There are hundreds of these, each designed to work well with certain types of goals or data.

Artificial Intelligence (AI)
General term for computer systems that show human-like intelligence. “General AI”, or a system that thinks and reasons like a human in many different settings, is still fiction. The current technology is “narrow AI”, which performs as well as or better than humans at a single very specific task.

Artificial neural network
A machine learning technique first introduced in the 1950s. Makes predictions using a model loosely based on the human brain.

Big Data
Originally, a term for data sets that were too large to be tackled with traditional data processing approaches. Now, this frequently refers to the use of predictive analytics to find patterns in such data sets and make predictions about the future.

Classifier
A machine learning algorithm that divides inputs into one of several output classes. Classifiers can be binary (yes/no decisions) or multiclass (choosing one of a larger set).

Data mining
The overall process of analysing large data sets to obtain understandable structure and knowledge.

Deep Learning
An artifical neural network with many layers.

Feedback loop
The interaction loop between the user and the machine learning algorithm. The algorithm makes predictions from data, which are

shown in the UI, causing the user to take an action. This action creates a new piece of data for the algorithm to learn from.

`Intelligence Augmentation (IA)`
The application of digital technology to amplify and improve human capabilities including thinking, analysis and planning. Unlike AI, IA is not intended to work autonomously but rather in close cooperation with and under human direction.

`Machine Learning (ML)`
Collective term for algorithms that learn from data. These algorithms are given example data, and some kind of goal, and learn their own rules for accomplishing the goal. ML is different from regular programming, where a specific set of rules is written by hand by the human programmer.

`Model`
The set of rules learned by a machine learning algorithm from the training data.

`Negative example`
In the training data for a binary classifier, a data point that matches the “no” class. E.g in the “is this a cat” classifier, a picture of something other than a cat.

`Pattern Recognition`
An alternative term for machine learning.

`Positive example`
In the training data for a binary classifier, a data point that matches the “yes” class. E.g in the “is this a cat” classifier, a picture of a cat.

`Predictive analytics`
The use of machine learning and statistics to study historical data in order to make predictions about unknown future events.

`Training data`
The historical examples studied by a machine learning algorithm to learn to perform a given task.


THE INTELLIGENCE
AUGMENTATION
DESIGN TOOLKIT
Futurice
UNCOVER STRUCTURE
Discover groups.
e.g. user segments, related
products, similar images

Toolkit materials

This part of the booklet briefly describes the contents of the IA design toolkit. It will guide you on using the materials to create a smart service concept.

The tools guide you through the following steps as you build up your idea:

- Identifying your customer and their need
- Mapping the customer journey
- Identifying places where machine learning can add value
- Choosing the machine learning features
- Reasoning about errors the ML might make
- Discussing important issues related to adding ML to the service
- Planning for potential major failures that could occur

The kit includes canvases, journey maps and card decks which are designed to structure your thinking and provide inspiration and points for consideration as you work. Over the next few pages we'll discuss the tools include in the Creative Commons version of the kit, available for free download online.

For commercial purposes, we can also offer an enhanced toolkit tailored for your organization's needs. Our experts in design, business consulting, and data science will facilitate a service creation workshop, supporting your team with their years of experience on emerging technologies and design. These workshops are targeted to the problems facing your business, and we can go in-depth on topics such as machine learning, GDPR, and biometrics. If you're interested, you can find local contacts for each Futurice office at the end of this booklet.

Who is the customer?

It's essential to design with your customers in mind. Who is your service or product meant for? For the shopping mall context used in Futurice organised workshops, we provide you customer segment cards based on Fonecta market research. If you are using the free version of the tool, we provide a template for making your own customer segment cards. Identify the segment(s) you are designing for and identify what need/problem your service addresses.

If you are working with your own project, we recommend you do a customer research with your clients or possible new clients. Interview your customers, do observations, ask people about their life. Spot problems and needs they might have. Get a picture of your customer's life and find the pain points where your service could add value for them.

Drawing the customer journey on a map

Once we have a customer, we can start mapping out their user journey. There are a number of tools for this: you might use our example map of a shopping mall, a map of your own creation, or the customer journey canvas used in our full-day workshops. These tools all have a common purpose: to help you find problems worth solving. Which tool is best for you depends on the context: think of the setting where you want to create the service, then choose a tool that lets you map the customer's journey. If you are creating your first concept ever, or just want to try out the toolkit, we recommend using the shopping mall map.

The idea of the customer journey is to think about how the customer currently accomplishes their need now, and use that as a starting point to figure out how your service could improve the process. You should ask questions like:

- What does the customer do to fulfil their need and why?
- Where do they go?
- What kind of emotions might your customer feel during their journey? An emotional map helps you to identify what parts of the journey are important and/or difficult and need special attention.

- What kind of touchpoints could the customer encounter on the journey? These are places where the customer might interact with the service you create. Use the touchpoint cards to help you.
- What is the real problem that needs to be solved? Is the existing journey even necessary?

Ideating smart service concepts

Based on the journey you have mapped out, you can start brainstorming possible service ideas. Remember, the goal is to help the customer fulfil their need by addressing pain points. For each idea, try to answer:

- How would this service improve the customer experience?
- Which pain points does this service address?
- Which touchpoints will the customer use to interact with the service?
- Can machine learning add value at one or more of the touchpoints? Use the interaction cards to remind you what machine learning can do.
- How will the customer discover what the service does?
- What happens after the customer uses your service?

Cards

The toolkit contains several card decks: touchpoints, machine learning interaction cards, and unexpected bug cards. The cards are designed to act as inspirational tools. They structure and prompt your thinking as you create your concepts.

Touchpoint cards

Touchpoint cards represent channels where machine learning might be added. They give you a set of technologies and physical elements to design your service around. Use these cards to identify what the customer encounters on the journey - these are


STORE PERSONNEL
ARTIFICIAL ASSISTANT
FACE RECOGNITION
INTERACTIVE SCREEN
TARGETED MARKETING CAMPAIGN
Entering the mall
Looking for a shop
Wondering around the shop

possible interaction points for the service. The deck includes a range of items, from everyday ones like mobile phones to futuristic ones like robots and smart shopping baskets. The set was originally designed for the context of a shopping mall, and it's still evolving. Feel free to draw new touchpoints if they are missing from the deck.

Machine learning interaction cards

Remember the four machine learning interaction categories from the beginning of this booklet? They described the common applications of machine learning. We have the same categories in this card deck. Using these cards, you can identify possible applications where machine learning might add value to your users. With machine learning, you might:

- Predict something about the future.
- Personalize the behaviour of a system to specific users or groups of users.
- Recognise information from input signals
- Uncover structure and identify interesting patterns and information in data.

Unexpected bug cards

Unexpected bug cards describe common failure cases that affect machine learning systems. With the help of these cards you can identify which failures might impact your service, and design to minimise those risks. Pick a card and spend a moment thinking how the bug could affect your design. Think about the kinds of strategies you could employ in order to cope with the bugs or prevent their appearance.

Note: not all bugs are relevant for all services. The cards help you think through different scenarios and pay attention to those which might cause your service trouble.

Canvases

The toolkit contains two canvases. As you flesh out your concept, fill in your thoughts on the Smart Service Canvas. This will help you to keep track of the most important issues you discuss. The Confusion Matrix Canvas will help you design for machine learning errors that might occur.

Smart Service Storyline

This is the canvas that helps you to structure your smart service concept. There are several versions of the canvas: the basic one available in the free version, and tailored ones we have used in various workshops. However, in all versions the basic idea is the same: the canvas has several boxes to write down answers to important questions, e.g.

- What does your service do?
- Who is your service for?
- Do you have a particular customer segment? Or is your service created according to a special need?
- What kind of problems will your service solve? Remember, not all problems are machine learning problems.

If you need machine learning, what is the main channel or touchpoint, you have chosen for enabling it?

- What value will machine learning bring? Why would your customer use your service? What would your company gain from using machine learning in the chosen touchpoint? How will you utilize the data you collect?
- How will your system learn from its interactions with users?
- Should the machine learning be explainable?
- What kinds of bugs or biases could affect your service?


CONFUSION MATRIX
Service idea
Magic Mirror
Machine output/predict
Negative
Positive
Service recommends me good moustache wax
I don't get to see the wax I want
Reality/Customer reaction
Positive
Negative
Service recommends me a wax I hate

Confusion Matrix Canvas

Remember the Netflix example in the beginning of this booklet? That was a confusion matrix. This canvas helps you in designing for failures in your machine learning. It can be used as a communication tool between designers and data scientists, giving hints on which kind of errors should have a higher cost when training models. It can also be used to ensure you have good fallback strategies in the service so that users can recover from the errors.

To fill out the matrix, figure out what a single decision looks like in your machine learning, and try to identify the positive and negative cases. For example, in a recommender system, a single decision is a prediction about whether a single user will like a piece of content (positive) or not (negative). In a fraud detection system, one decision is a prediction that a single transaction is fraudulent (positive) or not (negative).

By comparing the machine learning output to reality, you get four scenarios:

- True positive: ML makes a positive decision that the user agrees is correct
- True negative: ML makes a negative decision that the user agrees is correct
- False positive: ML makes a positive decision, but the user thinks it shouldn't have
- False negative: ML makes a negative decision, but the user thinks it should have been positive

For your concept, figure out what these situations look like. Then, try to answer these questions:

- What is the customer's likely reaction in each scenario? How will they feel when different kinds of errors happen?
- If an error occurs, how can the user recover from it? Is this process different between false positives and negatives?

- Are there any financial risks or privacy breaches that could occur in error situations?
- Could either kind of error cause a significant effect on the user's life?

Based on these answers, assign an overall cost to the different errors. You may even realise that the risks are too severe, and that the concept needs to be rethought. If you decide to keep developing the concept, make sure the fallback behaviour is well designed for each error type.

In our workshops we only concentrate on one application of machine learning at a single touch point, in order to teach the concepts. When designing a real service, you may end up with several different machine learning components. Pay attention to the whole customer journey, and fill in confusion matrices for each component. You might need to fill in several matrices for the same component because different use cases and user segments might not have the same logic.

By now you have enough information to start using the toolkit and create some service concepts. We hope you enjoy the journey and consider this as a good start towards designing the services of the future.

The latest version of this toolkit and booklet can always be found at our website, iadesignkit.com

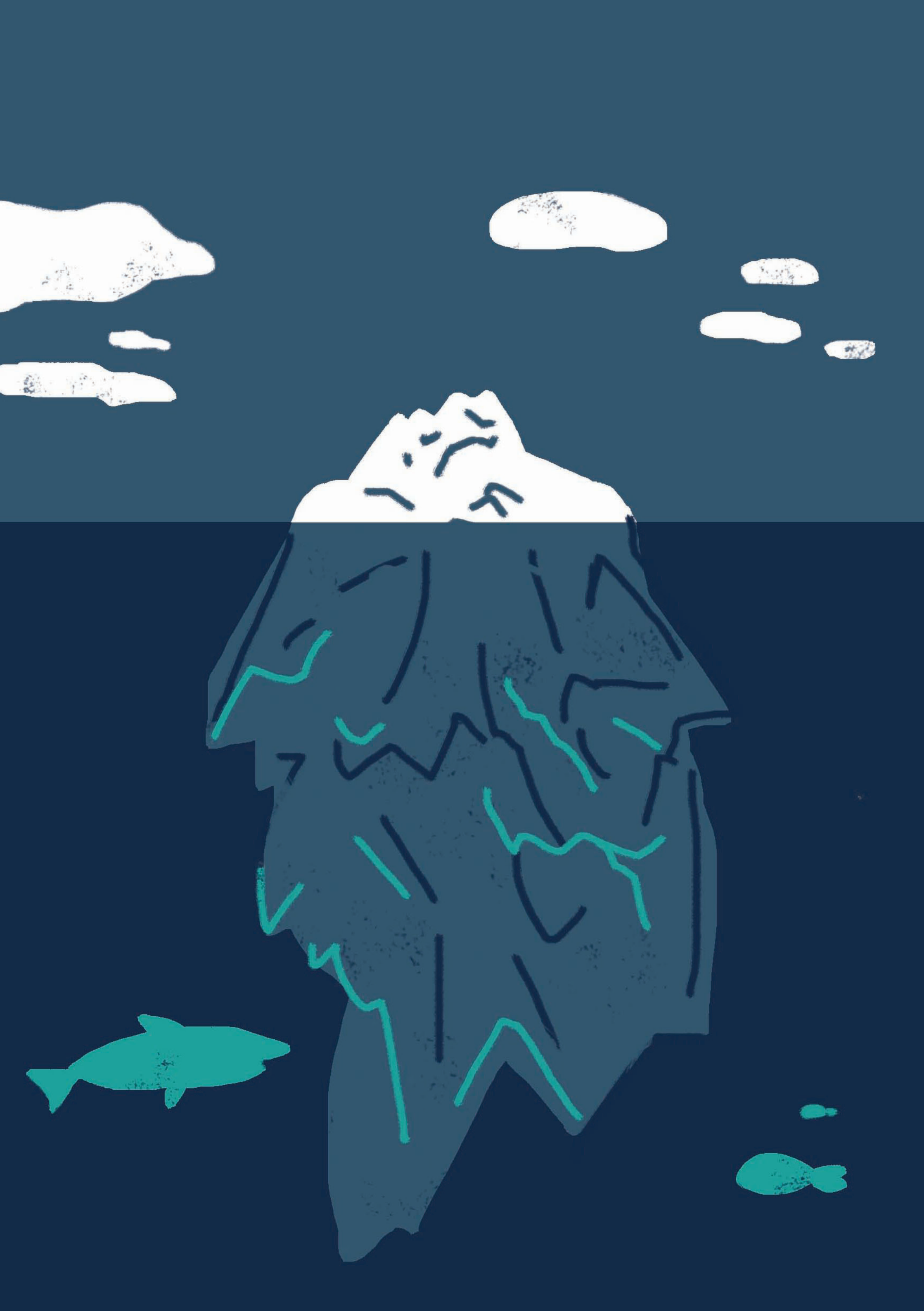
Why was this product recommended for me?
Bias
Always recommends the same wax

Great power, great responsibility

Machine learning is a powerful new tool in digital service creation, and it's only going to become more common. Like any tool, it has the potential to be misused, and the consequences can be severe. Examples are already appearing: from facial recognition being used to identify and target protesters on social media, to hyper-specific ad targeting to influence voters in elections. As designers and technologists, we face a growing responsibility to think about the ethics of the systems we design and build.

Sometimes the ethical implications are clear - say, in the case of autonomous weapons - but even seemingly benign systems can cause problems. Object tagging in an online photo library can assign racist labels, while algorithm curated feeds on social media can create echo chambers and increase political divisions. It is part of the designer's responsibility to think about how bias could manifest in a service, and collect data and design safeguards to try and prevent this.

At Futurice, we have an optimistic vision of the future, where machine learning supports and enhances our capabilities in many fields. That vision can only be achieved by close collaboration between designers, data scientists, developers and business experts. It can only be achieved if no one loses sight of the ethical responsibility that comes with creating these services. We hope our toolkit inspires you to design great things, and reminds you to think through the consequences of your designs. You have power. Use it wisely.



Want to hear more?

We’ve tried to capture some of the key insights needed when designing digital services that use machine learning, but there are many more that we don’t have space for.

We offer half- and full-day workshops dedicated to this topic, as well as consultation on your specific needs.

To learn more about these services, contact us:

Helsinki, Finland
annina.antinranta@futurice.com

Tampere, Finland
suvi.helen@futurice.com

Munich, Germany
yannick.engel@futurice.com

Stockholm, Sweden
ayca.ozkan@futurice.com

This Booklet is part of a workshop kit. The Intelligence Augmentation Design Toolkit was developed by data scientist Daryl Weir, tech wizard Paul Houghton, service designer Annina Antinranta and visual designer Alena Parshina.

Booklet text: Daryl Weir, Annina Antinranta, Paul Houghton, Arttu Tolonen. Visuals: Alena Parshina, photos: Daniel Morales, layout: Annina Antinranta.

futurice

iadesignkit.com