

# ML and Society

Feb 27, 2023

# Passphrase for today: Grace Wahba



Grace Goldsmith Wahba

It is easy (and accurate) to say that Grace Wahba is this country's most eminent female statistician, but it is also misleading: Grace is in the top few among *all* American statisticians, and is in fact one of our most important applied mathematicians. She was also the first female faculty member in the Department of Statistics at UW–Madison. Supervising the theses of 39 graduate students from four continents, her career there lasted 51 years, ending with **retirement in August 2018**.

Grace almost owns the word “spline” as it is used in statistics; her 1990 monograph *Spline Models for Observational Data* has been a scientific best-seller. It is based on a series of fundamental papers written with various co-authors in the preceding years. Of these, the most influential might be 1979's “Generalized cross-validation as a method of choosing a good ridge parameter”, with Gene Golub and Michael Heath, which introduced the **GCV (generalized cross validation) criterion**. The hallmark of Grace Wahba's work is a combination of high-powered mathematics, often involving functional analysis ideas such as reproducing kernel Hilbert spaces, but with the practical problems of real data analyses kept firmly in view. The very best statisticians always turn out to be good scientists as well as

# Reminder about CSE 199 IP

**FEATURING:**



**THE IMPOSSIBLE PROJECT**

**MAKING COMPUTING ANTIRACIST**

**FINALE EVENT**

**FEBRUARY 27TH**

**5PM, NSC 225**

**VIRTUAL REGISTRATION**



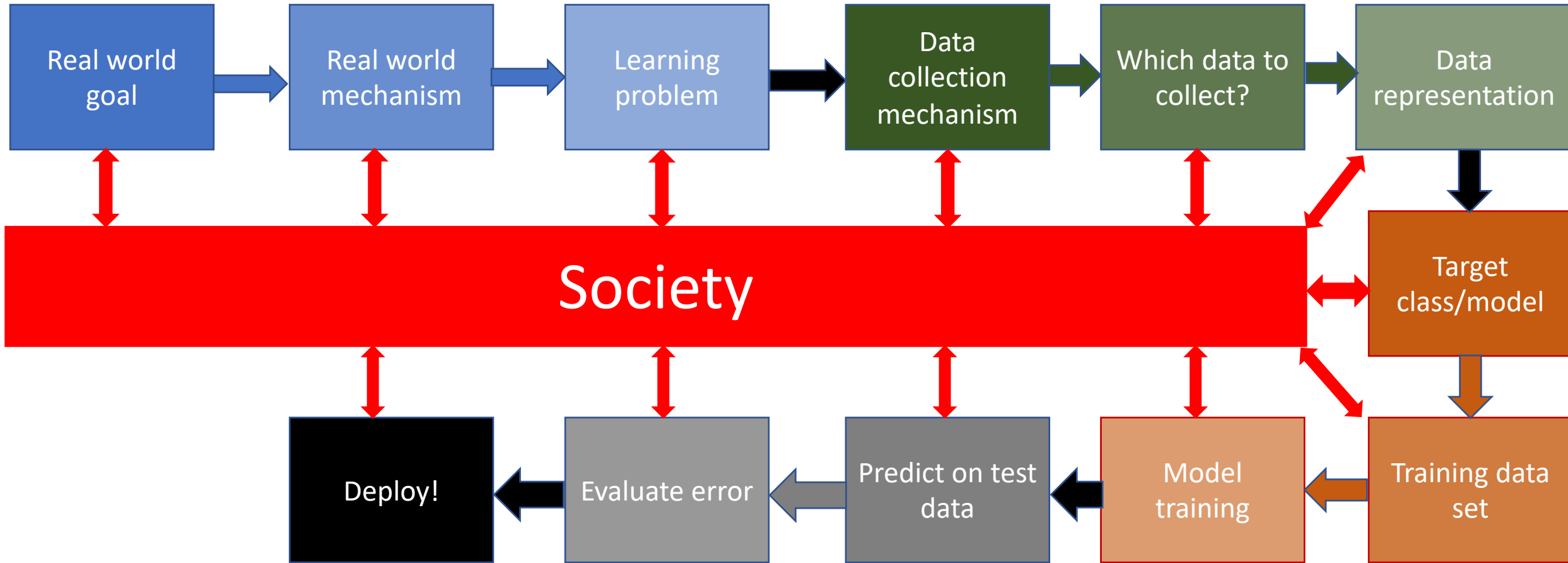
2. For **200 bonus points**, you can attend the **CSE 199 Impossible Project Finale Event** in person at 5pm on February 27th in NSC 2225. You'll need to attend the entire event (~75 minutes), checking in with Kenny or Atri at the beginning and the end. You can receive **50 additional bonus points** for asking a question during the Q&A period as well.

# Sit with your team!

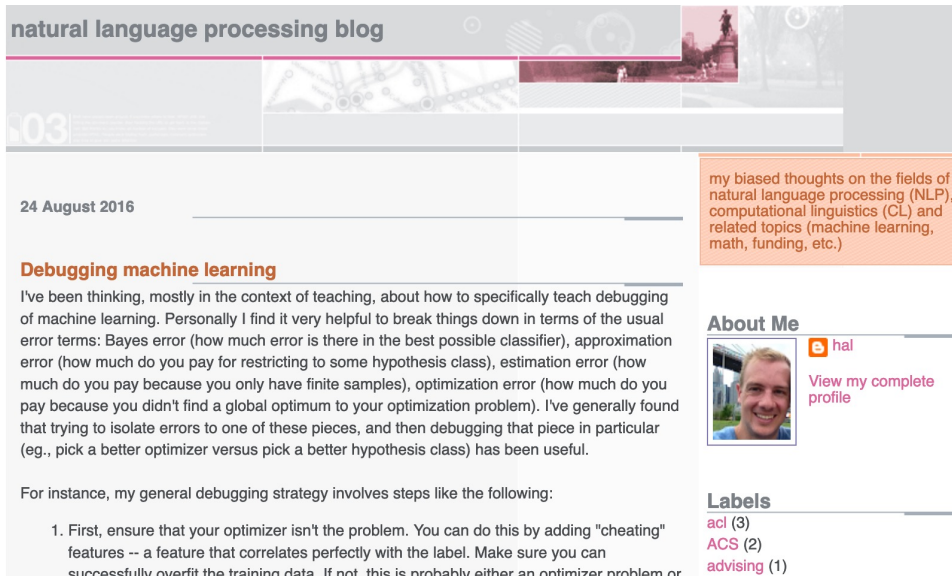
Team 1	Afzal	Cole	Navid	Tim	
Team 2	Aishwarya	Herman	Mads	Melvin	
Team 3	Daphkar	Juliana	Ibtida	Monica	
Team 4	Joe	Ken	Vedant	Zach	
Team 5	Chaitanya	Evan	Hitesh	Sushanth	
Team 6	Hannah	Harinee	Gabriella	Suradhya	
Team 7	Alex	Connor	Gopi	Shane	Thanh
Team 8	Aditi	Connor	Jason	Mitali	
Team 9	Botsalano	Niharika	Vedang	Yunmei	
Team 10	Dhiraj	Frank	Kashyap	Michael	

Rage students in Green. ML+Soc students in black

# ML pipeline



# Three problems to consider



natural language processing blog

24 August 2016

### Debugging machine learning


I've been thinking, mostly in the context of teaching, about how to specifically teach debugging of machine learning. Personally I find it very helpful to break things down in terms of the usual error terms: Bayes error (how much error is there in the best possible classifier), approximation error (how much do you pay for restricting to some hypothesis class), estimation error (how much do you pay because you only have finite samples), optimization error (how much do you pay because you didn't find a global optimum to your optimization problem). I've generally found that trying to isolate errors to one of these pieces, and then debugging that piece in particular (eg., pick a better optimizer versus pick a better hypothesis class) has been useful.

For instance, my general debugging strategy involves steps like the following:

1. First, ensure that your optimizer isn't the problem. You can do this by adding "cheating" features -- a feature that correlates perfectly with the label. Make sure you can successfully overfit the training data. If not, this is probably either an optimizer problem or

my biased thoughts on the fields of natural language processing (NLP), computational linguistics (CL) and related topics (machine learning, math, funding, etc.)

**About Me**



hal

[View my complete profile](#)

**Labels**

- acl (3)
- ACS (2)
- advising (1)

Ad display example

### The Story of a Data Scientist

Jasmine is a data scientist working for a large university hospital. She works closely with the hospital management, working on multiple projects – analyzing trends in spending and medical procedure data and building statistical models to help the management and doctors gain a better insight into how redirecting resources to different patients and departments will affect spending, patient health and employee satisfaction.

One day, Jasmine is in a meeting with the management, where they discuss a newly established government program which provides the hospital with additional resources to help manage the health of patients with significant health needs. The program offers monthly meetings with a nutritionist, physical therapy, weekly, free-of-charge psychotherapy, as well as a personal program coordinator who is available 24/7 to support the patient and help them navigate their healthcare. The program was established to support elderly and diabetic patients, but it is at each hospital's discretion to select patients who will enter the program. There are 50 spots, for over 1,200 patients served by the hospital.

The management is very excited about the additional resources, but one of the senior doctors brings up that the selection of 50 most needy patients may be challenging. Should they select those with poorest health? Those who do not have relatives or spouses who help

Hospital trying to utilize new govt program

Third example: Prediction hate crime

# Real world goal

Real world  
goal

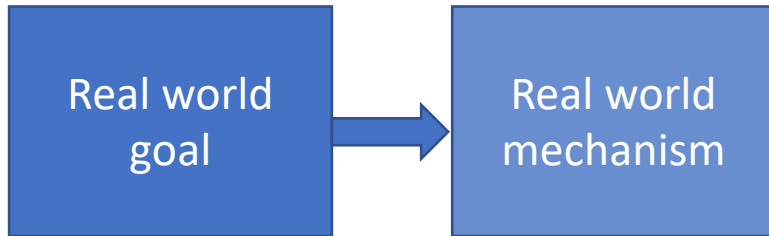
## Real world goal: Example 1

Your company wants to increase revenue. A majority of revenue for your company comes from facilitating online ads. Your group has to attain this high level goal.

## Real world goal: Example 2

Your hospital learns of a new government program that provides hospitals with additional resources to help manage health of patients with significant needs. The hospital management wants your hospital to utilize these funds since the hospital has been losing money in the last few quarters. However, the funds can only help a (relatively) small fraction of the patients in your hospital.

# Real world mechanism



## Real world mechanism: Example 1

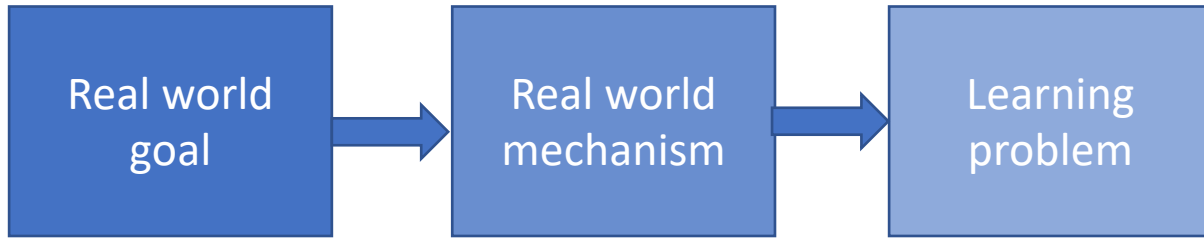
Since online ads make up a majority of the company's revenue your group decides to improve upon the ad display (with the hope that this can generate more revenue).

## Real world mechanism: Example 2

Here you get conflicting demands: the management wants to use the extra funds to cut spending (i.e. keep the current service at their current level) while doctors want to use the extra funds to supplement the existing services (i.e. add on to the existing services).



# Learning problem



## Learning problem: Example 1

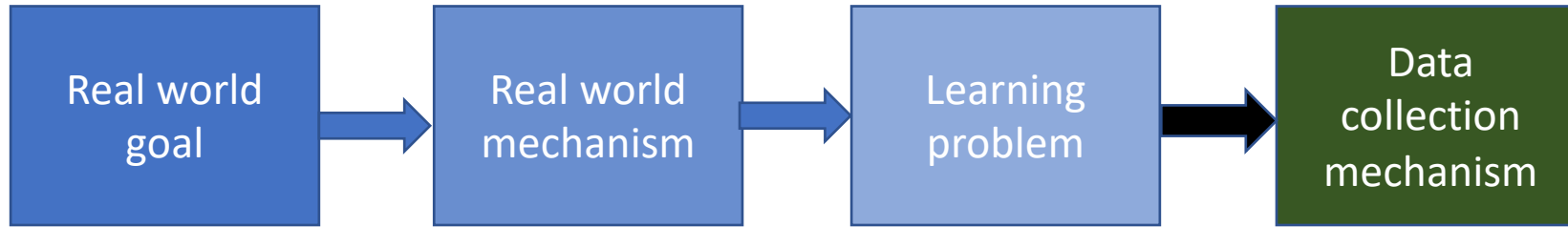
Your group decides to predict the [click through rate](#), which is a measure of the likelihood that a user will click on your ad. Based on these predictions, you will better place ads.

## Learning problem: Example 2

The doctors had their way so your group decides to predict the patients with most need so that they can be targeted with the supplementary practice.



# Data collection mechanism



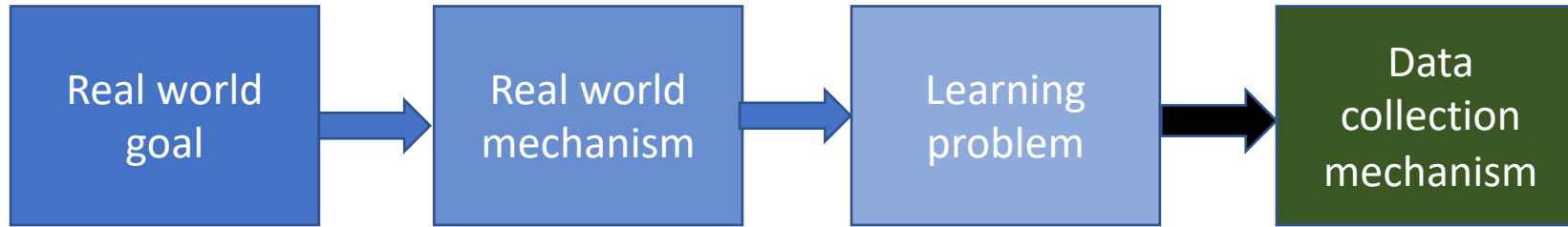
## Data collection mechanism: Example 1

Your group decides to log interactions with ads in the current system.

## Data collection mechanism: Example 2

Your group decides to use the existing patient electronic health records (which includes details of the current care the patients receive in your hospital but possibly other details).

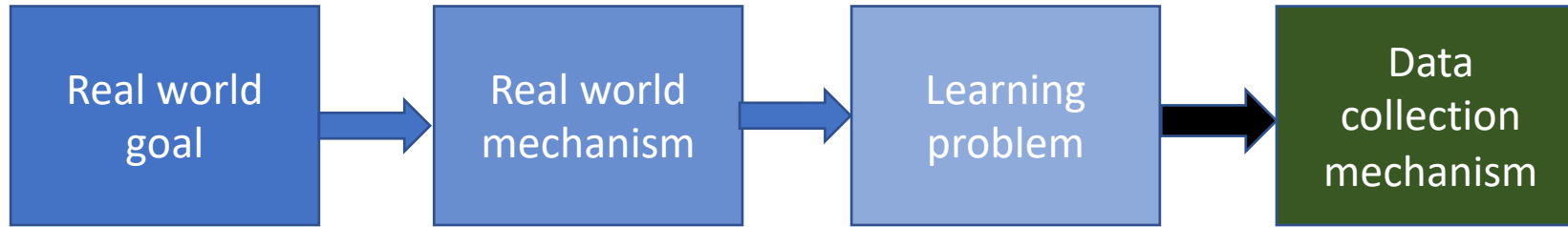
# Data collection mechanism: general thoughts



Concept/distribution drift

Privacy can be a concern

# Data collection mechanism: Data doesn't exist



Use 3<sup>rd</sup> party data brokers

## The Data Brokers So Powerful Even Facebook Bought Their Data - But They Got Me Wildly Wrong

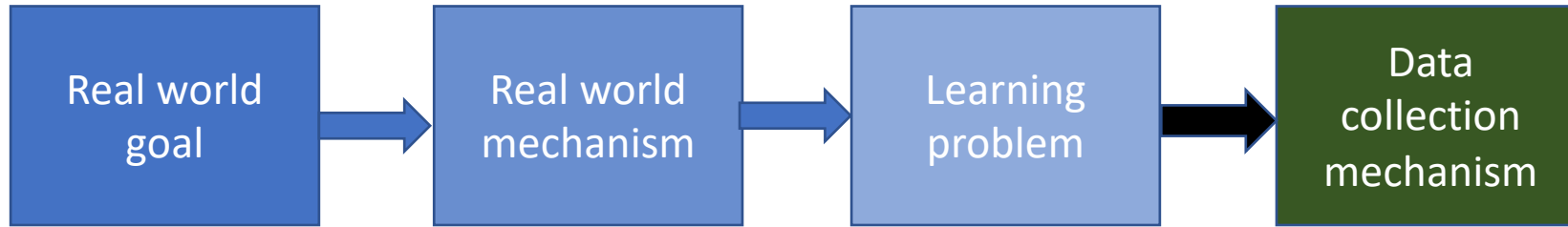


Kalev Leetaru Contributor @  
AI & Big Data

*I write about the broad intersection of data and society.*



# Data collection mechanism: Data doesn't exist



Run surveys



Products ▾

Solutions ▾

Resources ▾

Plans & Pricing

LOG IN

SIGN UP

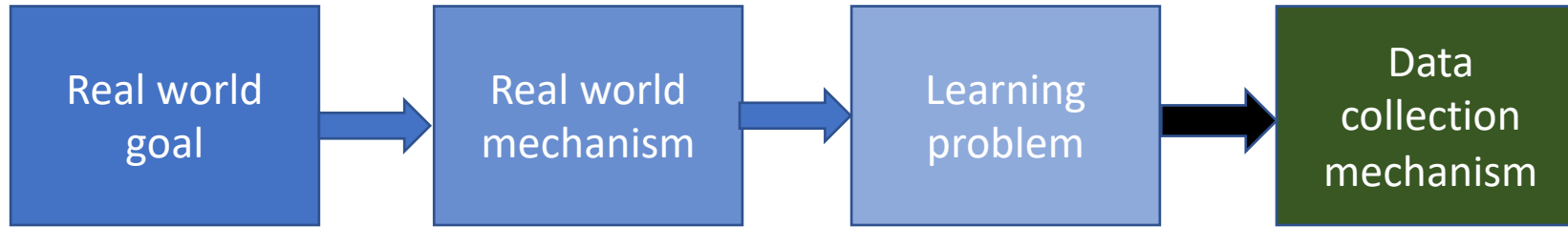
## Are my customers actually satisfied?

A global leader in survey software. 20 million questions answered daily.

GET STARTED

Potential issues?

# Data collection mechanism: Data doesn't exist



Collect data from smartphone



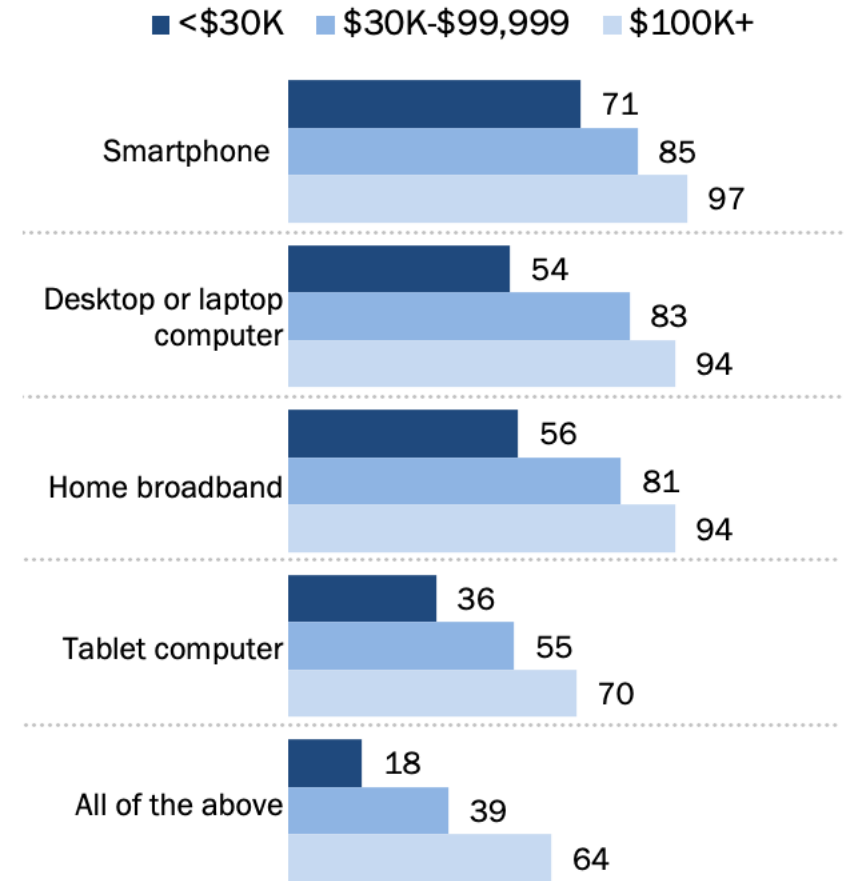
The screenshot shows the Street Bump website interface. At the top, there is a navigation bar with 'STREET BUMP' and 'About' on the left, and 'Sign In' on the right. The main content area features a heading 'Where's Street Bump being used?' followed by statistics: '549 trips, 37,016 bumps, 0 potholes filled, and 0 roadway problems identified'. Below this is a map of Truro, NS, with a callout indicating '99 bumps reported in 101-199 Curtis Dr, Truro, NS almost 6 years ago'. The map shows various streets and landmarks, with a 'STREET BUMP' icon on Curtis Dr. To the right of the map is a video player titled 'What's Street Bump?' with a play button and a '00:50' duration. Below the video is an 'App Store' download button and a 'Learn More' link. At the bottom of the page, there is a footer with the text 'Want to use Street Bump to improve your community? Contact Us'.

# The smartphone blind-spot

Many of us in CSE assumes that “everyone” has smartphones

## Lower-income Americans have lower levels of technology adoption

*% of U.S. adults who say they have the following ...*

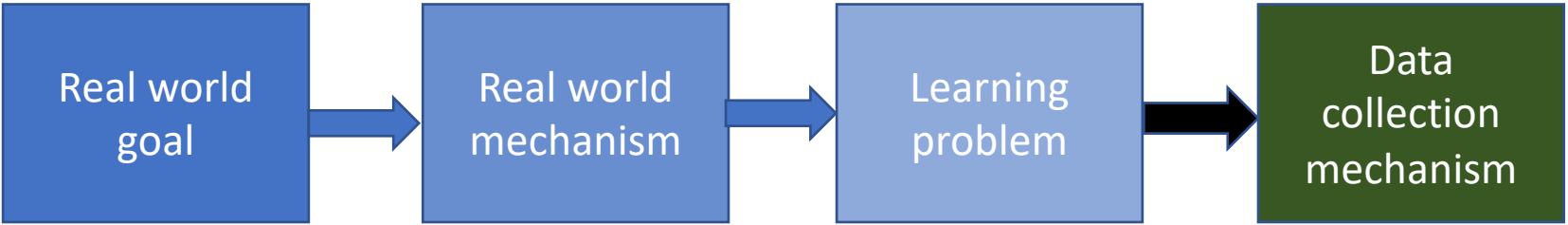


Note: Respondents who did not give an answer are not shown.  
Source: Survey conducted Jan. 8-Feb. 7, 2019.

**PEW RESEARCH CENTER**



# Data collection mechanism: Data doesn't exist



**MORAL MACHINE**

Online video games



Try our emotional AI  
(opens new tab)

DeepMoji



## Younger Americans and men are among the most likely to play video games

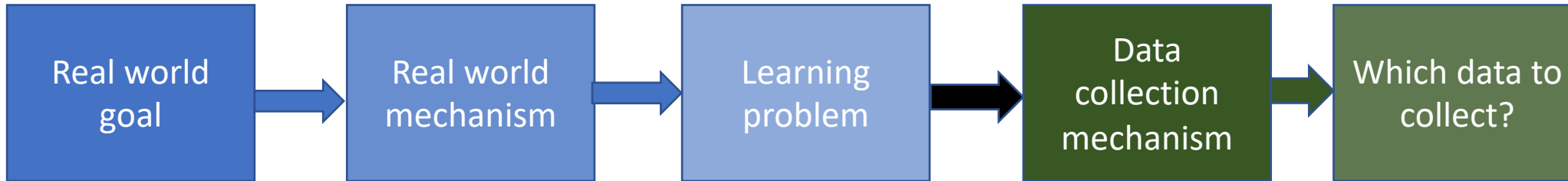
% of adults saying they often/sometimes play video games on a computer, TV, game console, or portable device like a cellphone

	Often	Sometimes	Net
Men	24	23	47
Women	19	21	39
White	21	20	41
Black	24	20	44
Hispanic	18	29	48
Ages 18-29	29	31	60
30-49	28	25	53
50-64	15	17	31
65+	11	13	24
High school or less	21	21	42
Some college	25	25	50
Bachelor's degree +	17	19	36

Note: Figures may not add to subtotals due to rounding. White and blacks include only non-Hispanics. Hispanics are of any race.  
Source: Survey of U.S. adults conducted March 13-27 and April 4-18, 2017.  
**PEW RESEARCH CENTER**



# Which data to collect?



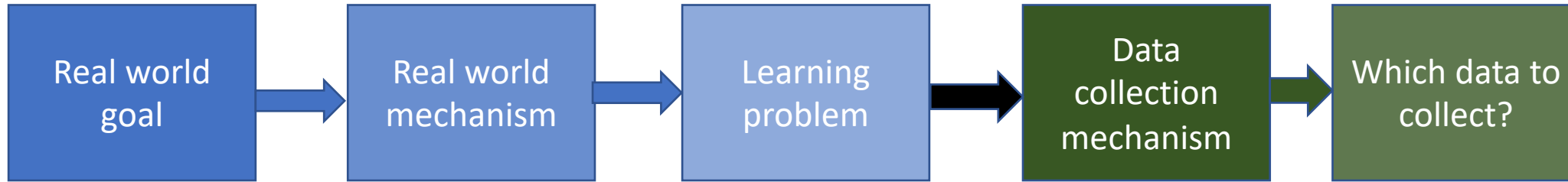
## Which data to collect?: Example 1

Even though you have access to the current system, you cannot log everything. This could be because e.g. sorting everything would need a lot of storage or perhaps if the system were to log every action it observes then just the act of logging everything can slow down the system (which is not desirable). For example, your group (as [Hal suggests](#)) decides to log queries (for which ads are generated), ads and clicks.

## Which data to collect?: Example 2

In this example, by restricting yourself to electronic health records, you are limiting yourself to what is logged into the electronic health records. One could e.g. try and use doctor's notes to glean more information but these are not necessarily standardized and it's not clear how to extract information from doctor's notes. Further, there have been [complaints from doctors on the usability of electronic health records](#), which raises issues about accuracy of data being collected. Finally, for the study that your group is planning will most probably need IRB approval from your hospital, which could in turn restrict which data can be collected/used for your system.

# Which data to collect?: General thoughts



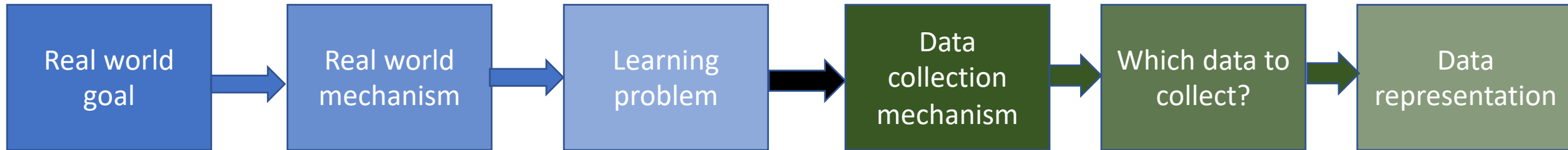
Expense might determine what gets collected

Time to finish a survey also has implications

Other restrictions, e.g. from an IRB



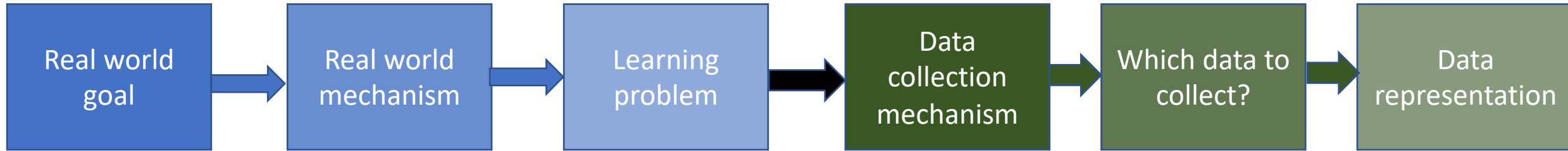
# Data representation




<https://www.history101.com/april-14-2003-the-human-genome-project-completed/>



# Data representation



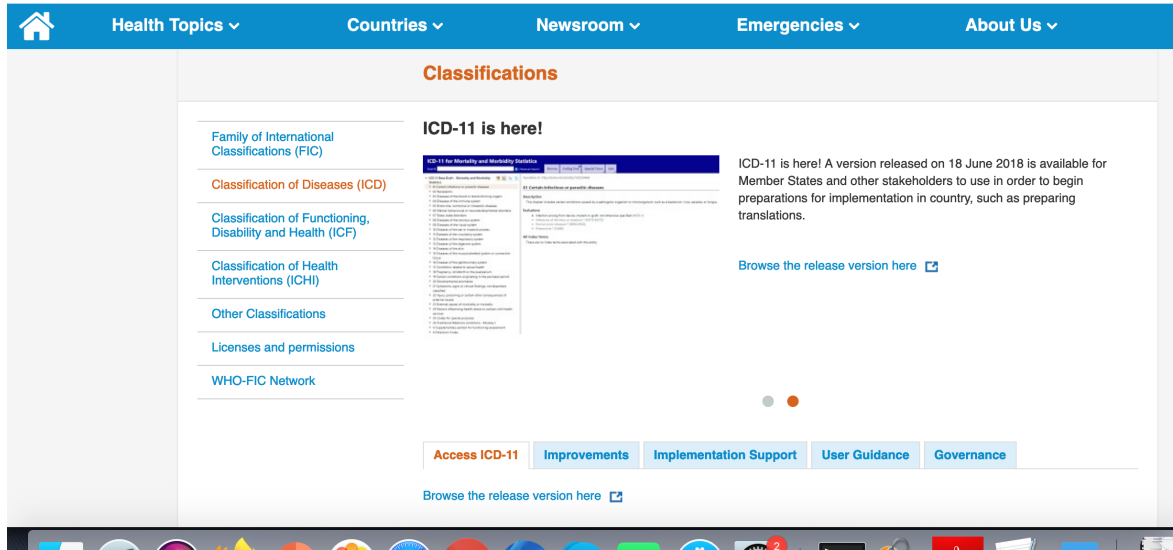
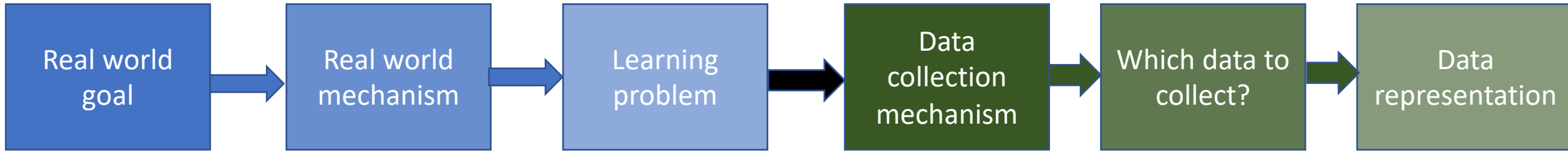
## Data representation: Example 1

Your group has zeroed in on query, ad and clicks. For the latter perhaps the most natural way to represent this to encode whether a user clicked on ad or not (so either + for clicked and – for not clicked or 1 for clicked and 0 for not clicked). The representation for query and the ad is not as straightforward. We could store the exact text for the query and the ad but that seems to indicate issues (e.g. what is you ad text are distinct strings but are essentially the "same" for human consumption or what if someone runs a query that has the same keywords as another query but in different order). To get around this issues by using the text as is, your group decides to use a representation that is more standard in natural language processing: [bag of words model](#) .

## Data representation: Example 2

In this case since your group is using the electronic health records, then the data representation is pretty much already fixed for your group. Perhaps one exception could be to represent the doctor's notes in the [bag of words model](#)  as above.

# Data representation: General thoughts

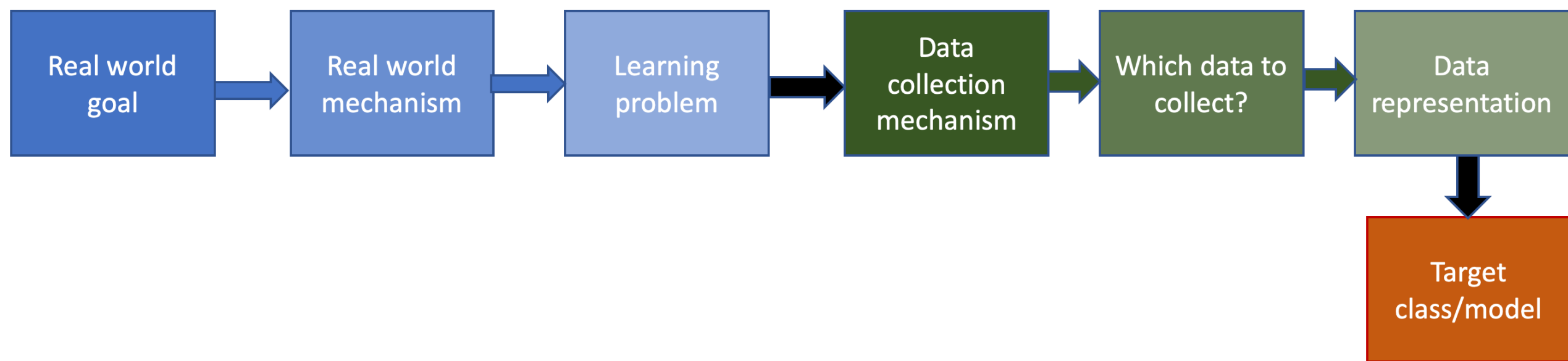


Categorical data

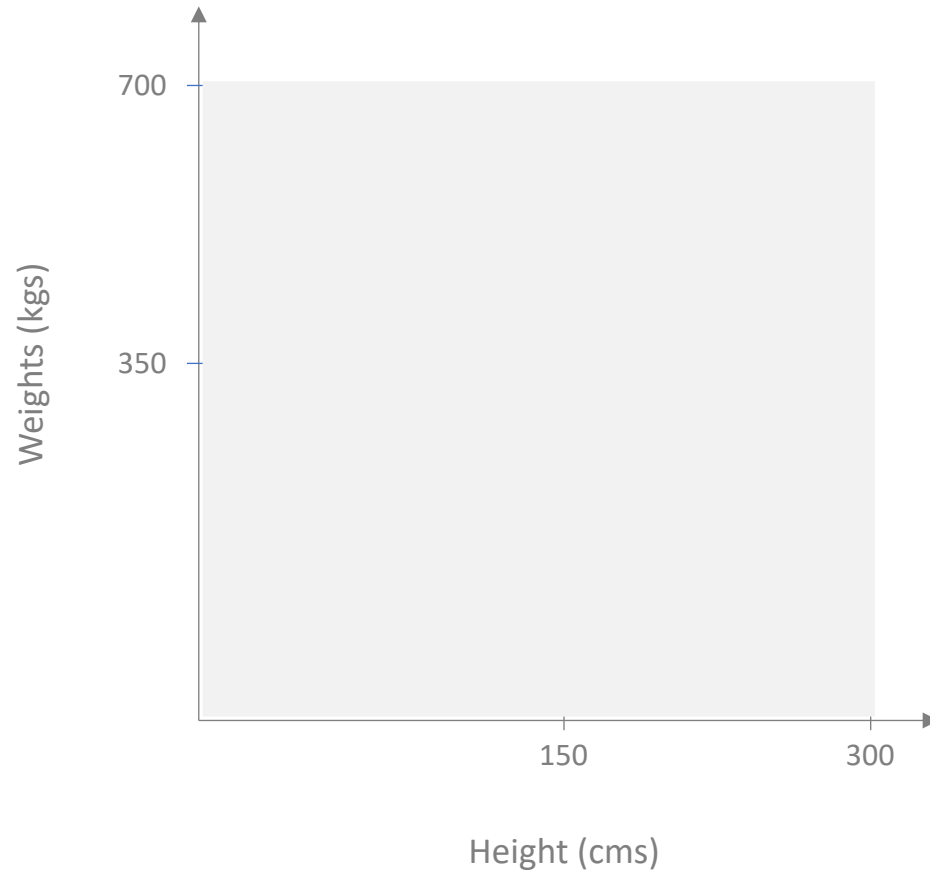




# ML model classes

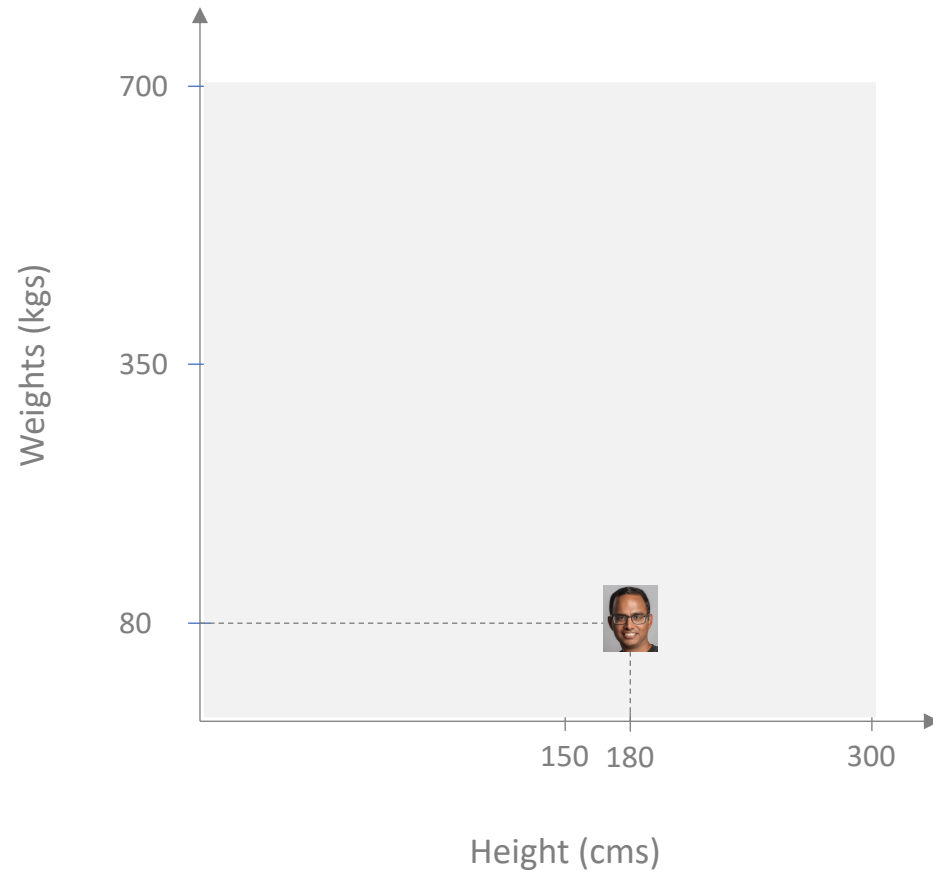


# Restrict to two input variables

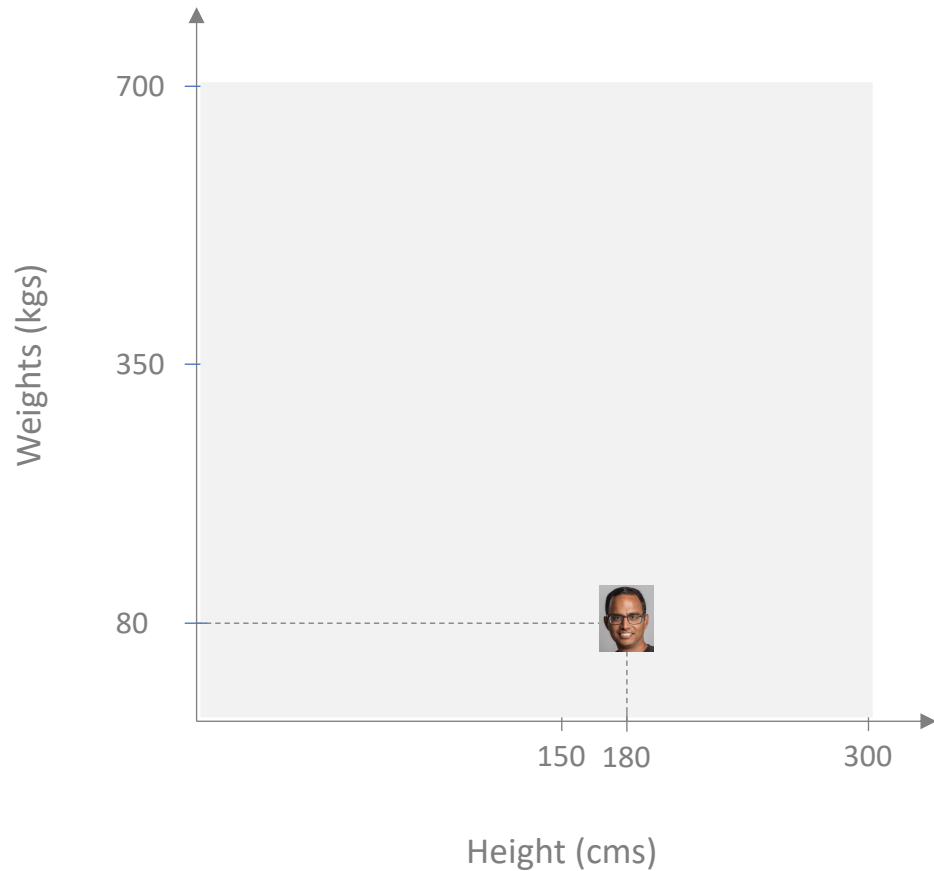


Predict risk of heart disease

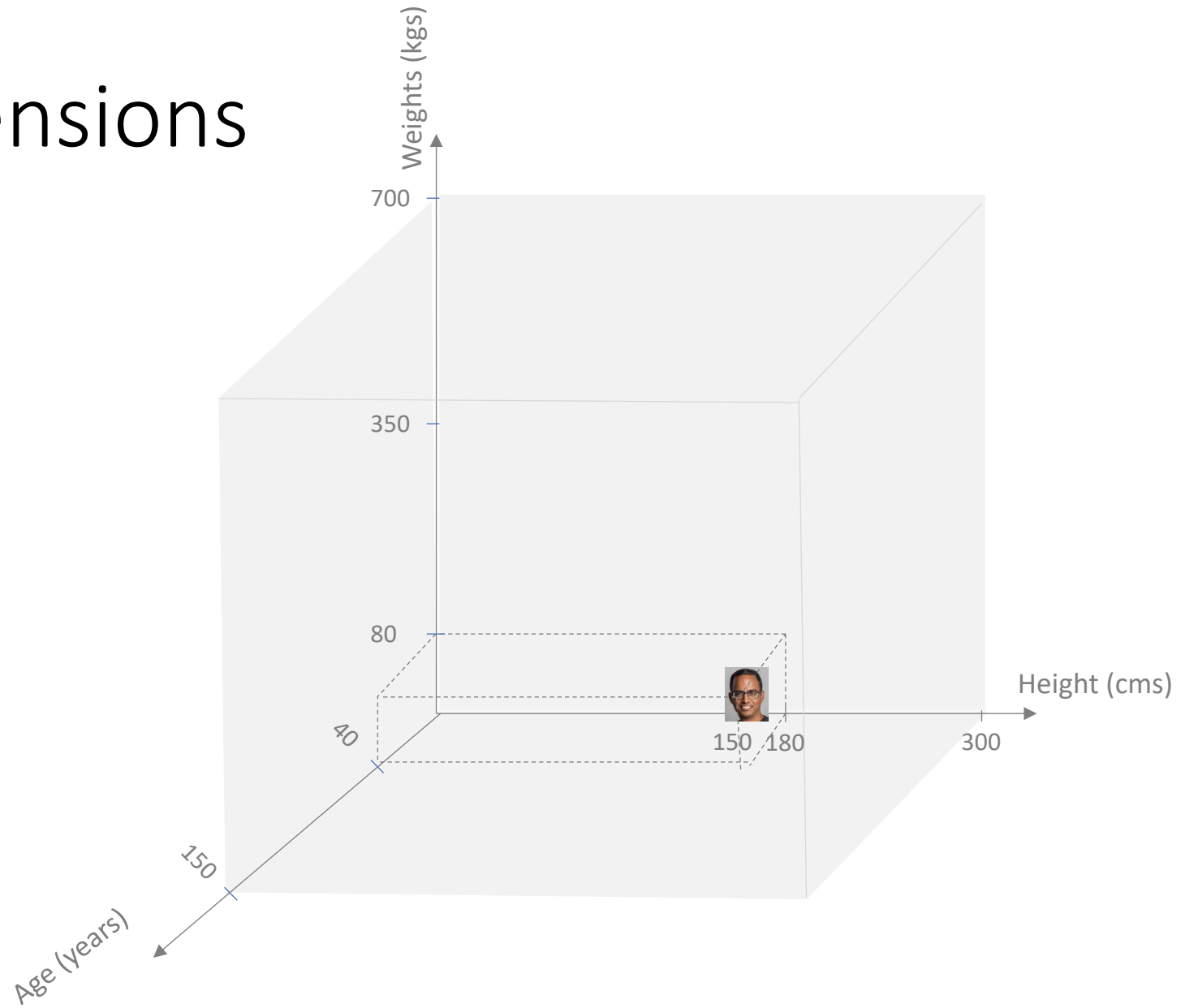
# For example...



# Does every human get their unique point?



# Need more dimensions





# Focus on binary classification

## Binary Classification is the name of the game

We had mentioned this in the passing [earlier](#), that in this course we will focus on binary classification: i.e. the target variables that we will consider in this course only take two values: typically we will call one value **positive** and the other **negative**. For example, in the running example in this notes of predicting the risk of heart disease, **positive** would mean high risk of getting heart disease and **negative** would mean low risk of heart disease.

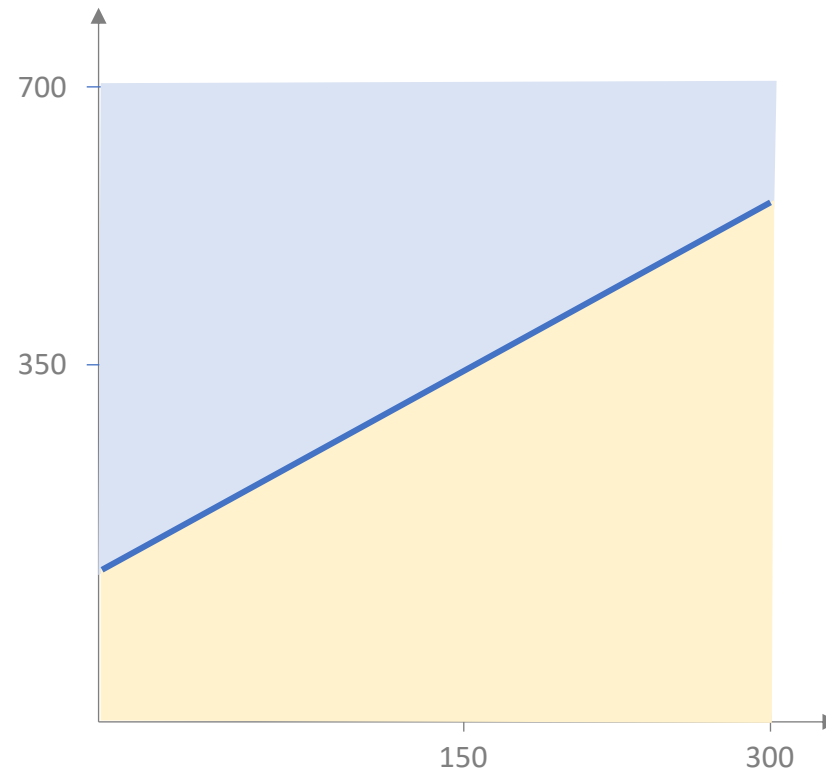
Of course, in many cases (including the heart disease risk prediction) it makes sense to have the target variable take many values (or even for the case when the target variable is inherently binary it makes sense to assign a probability of whether the target variable is positive-- in which case the target variable is a real number between 0 and 1 and thus takes on infinitely possible values).

We decided to focus on binary classification since it makes the exposition easier (including drawing figures of the kind y'all will see soon) but at the same time is complex enough to illustrate many of the technical details and nuances.



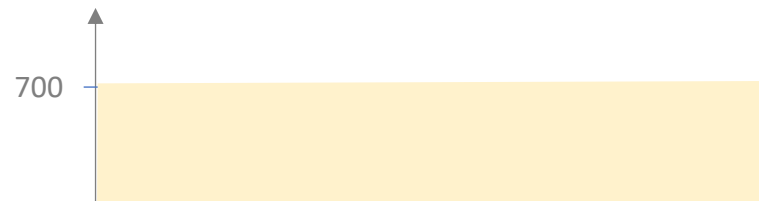
# What is a model?

A "curve"/function that labels every point either as positive (blue) or negative (yellow)



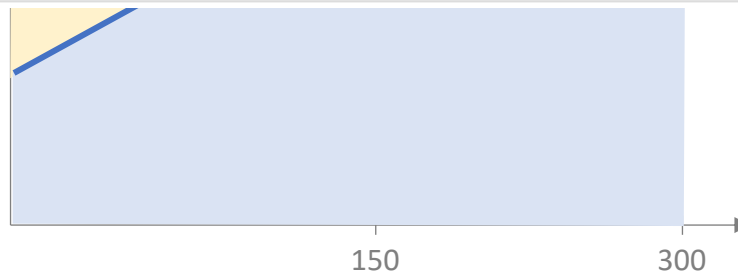
The line "defines" the model. Almost...

# The colors can be flipped!



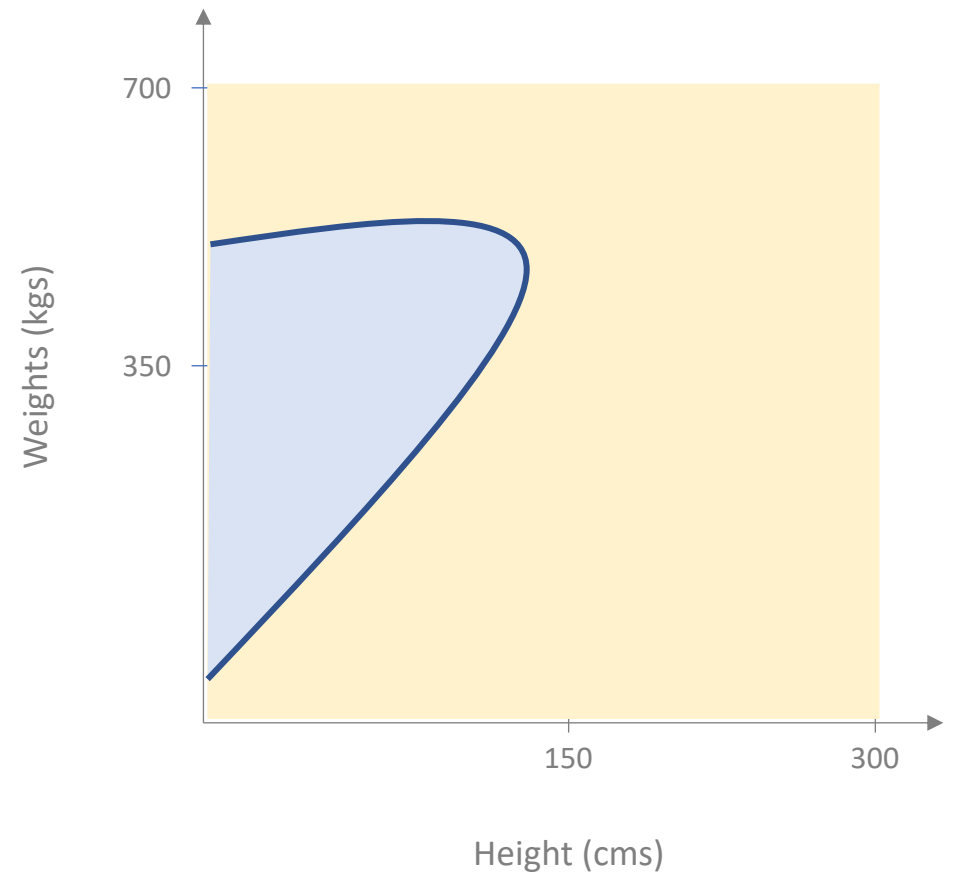
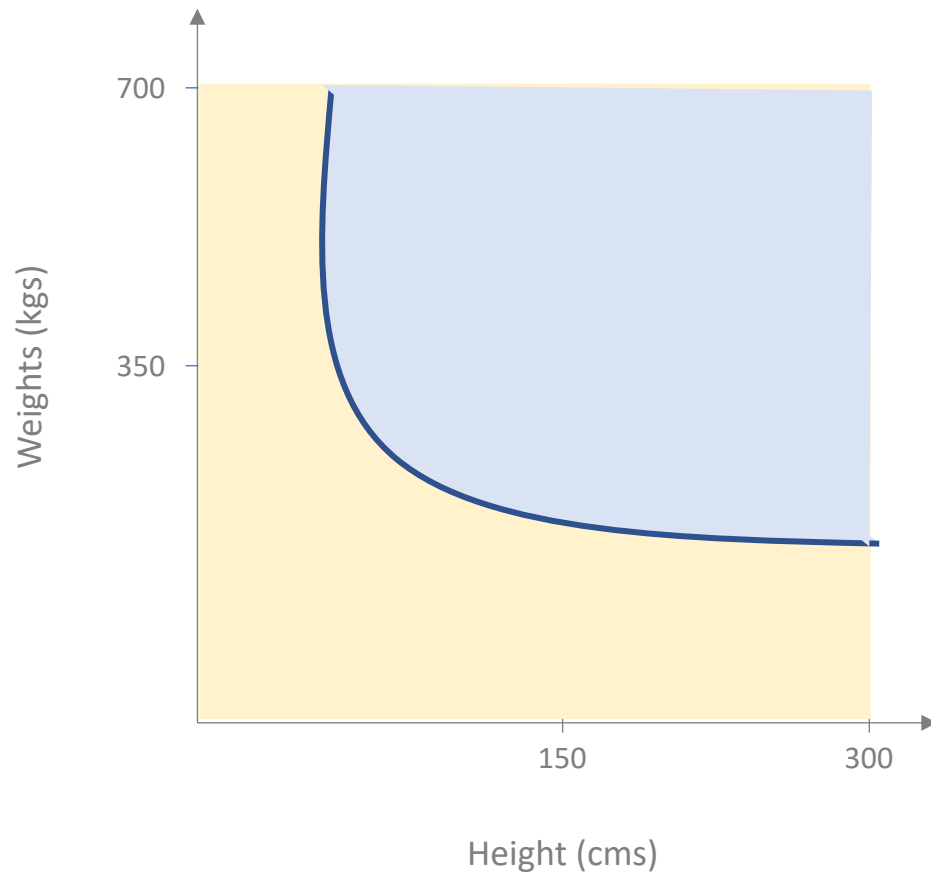
## Linear models

The models we considered above (as we have seen before) are called *linear models* (because you can literally draw a line to present them in 2D). In particular, in the case of two input variables, a linear model is **completely defined** once you specify the line as which of the two sides is the positive side (and the other side automatically becomes the negative side).

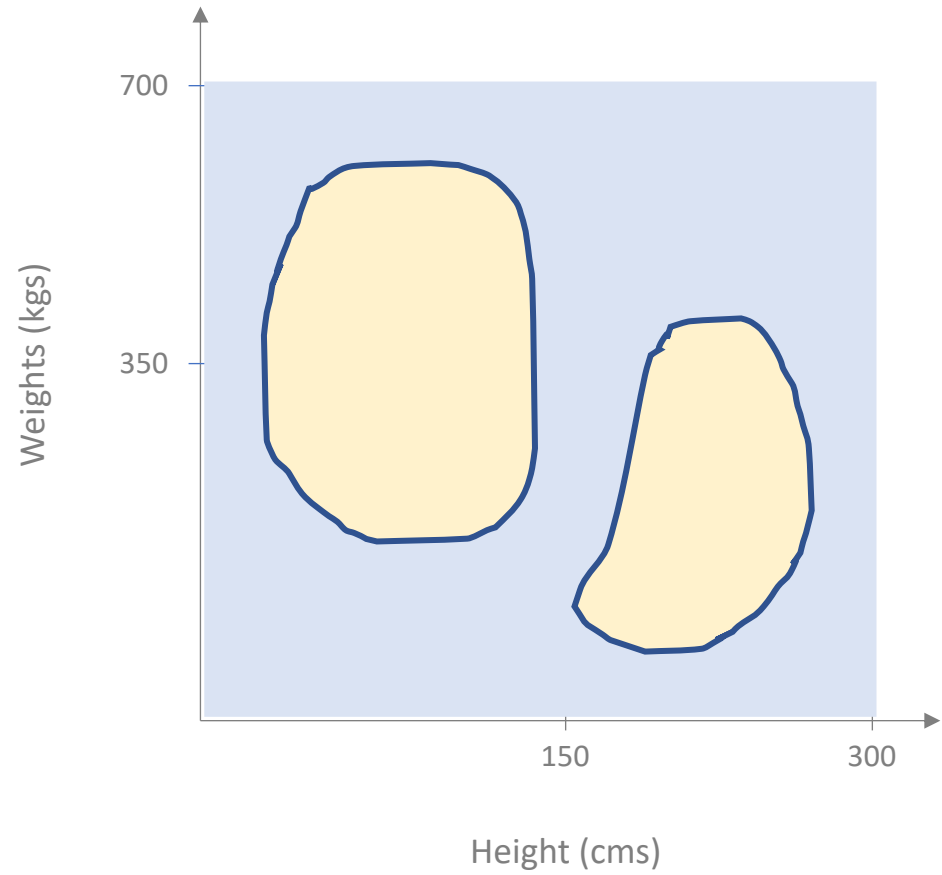
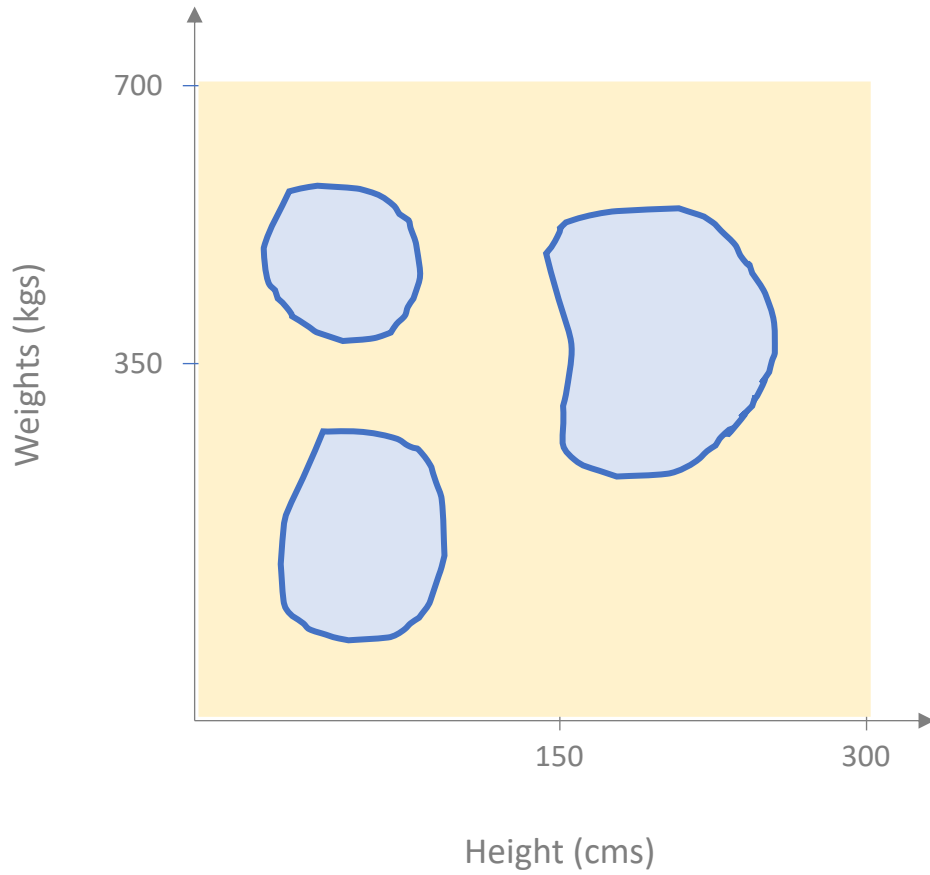


Height (cms)

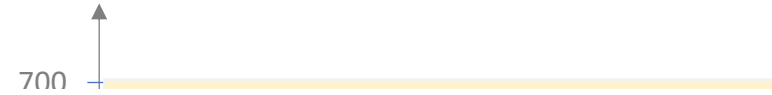
# Can go beyond linear models: why not?



It can get crazier...



# And crazier!



## The figures above are not of real models

In case y'all were wondering the non-linear models were all drawn by hand and do not correspond to a model that has been used by someone in real life. However, except for perhaps the last figure, they are approximations of models that are used in real-life.



## Models for binary classification

A model is a function (or a procedure if you will) that divides up the ambient space (in the running example it is the gray area in the first figure) into disjoint regions where each region is colored blue (for what the model considers the positive points) and yellow (for what the model considers to be negative points).

Height (cms)

Height (cms)



# Model class is independent from earlier steps

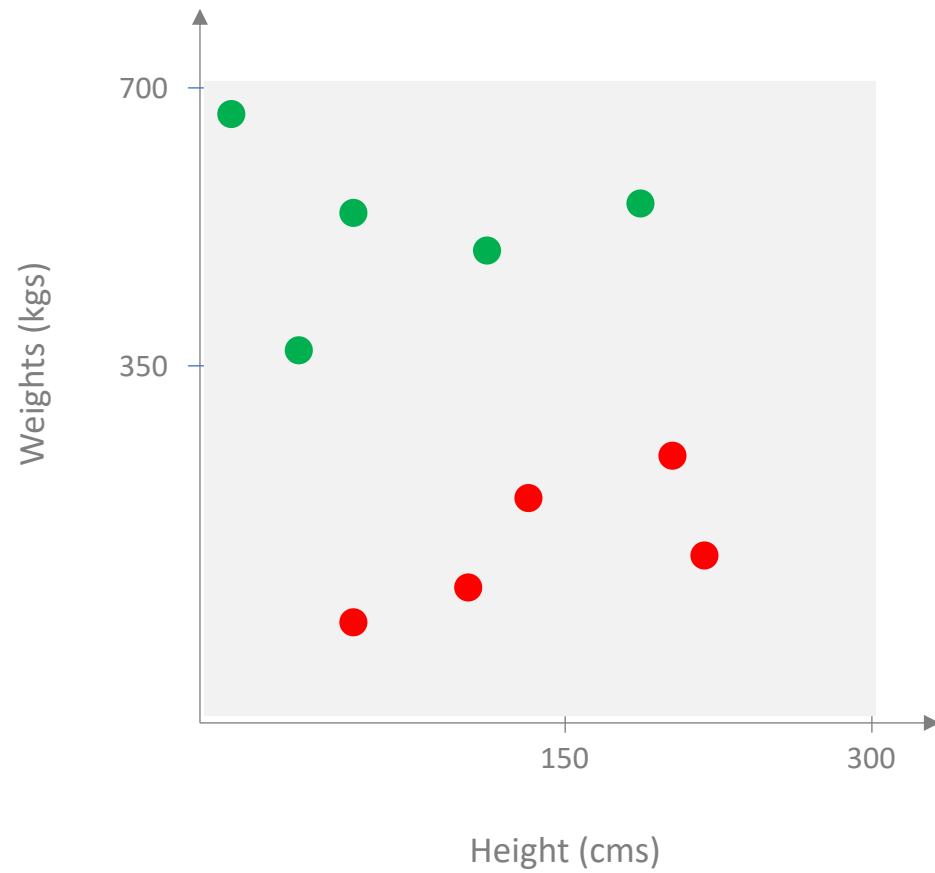
## Independence of model from problem/data

When we first looked at choosing the model class as part of our [walkthrough of the ML pipeline](#), by the time we looked at the linear model cartoon, we already had the labeled (red/green) points. However, so far we have not seen labeled points in our plots yet. Why so?

The fact that you can talk about a model *independent* of the training data is precisely the reason ML is so effective: i.e. traditionally it has been important for the models to be defined **independently of the problem/data** so that one can reason about them independently of the problem/data. This *abstraction* is what given ML its power. In particular, this abstraction allows one to use the same technology for different problems/data and allows for the wide-spread use of ML in many areas. Basically if one can do the seventh step of the ML pipeline (and beyond) independently of the first six steps, then one can use "off-the-shelf" ML systems for the seventh step and beyond of the ML pipeline and for a particular application, one can just concentrate on the problem-specific tasks. In particular, any advance in the non-problem specific steps of the ML pipeline can then be used for any problem where one has handled the first six steps.

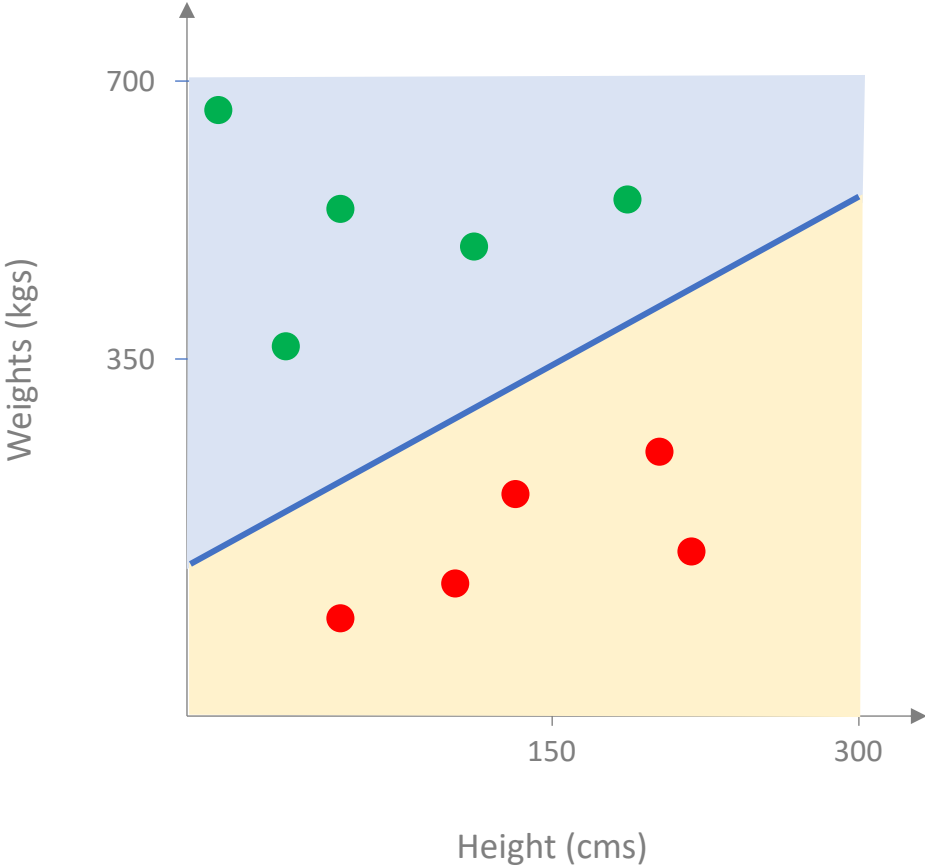
As mentioned above, the argument above is the *current/traditional* way ML systems are used. However, as we will see later in the course, such a clean separation between the first six steps of the ML pipeline and rest leads to situations with non-ideal outcome especially when it comes to questions of fairness or bias. The high-level reason for this is that even though we have drawn the ML pipeline as a sequential set of steps, in real life they are not so. Specifically, the fact that every step of the ML interacts with society means that the various are *not* independent of each other even though our flowchart might make it look otherwise.

# Is one model enough?

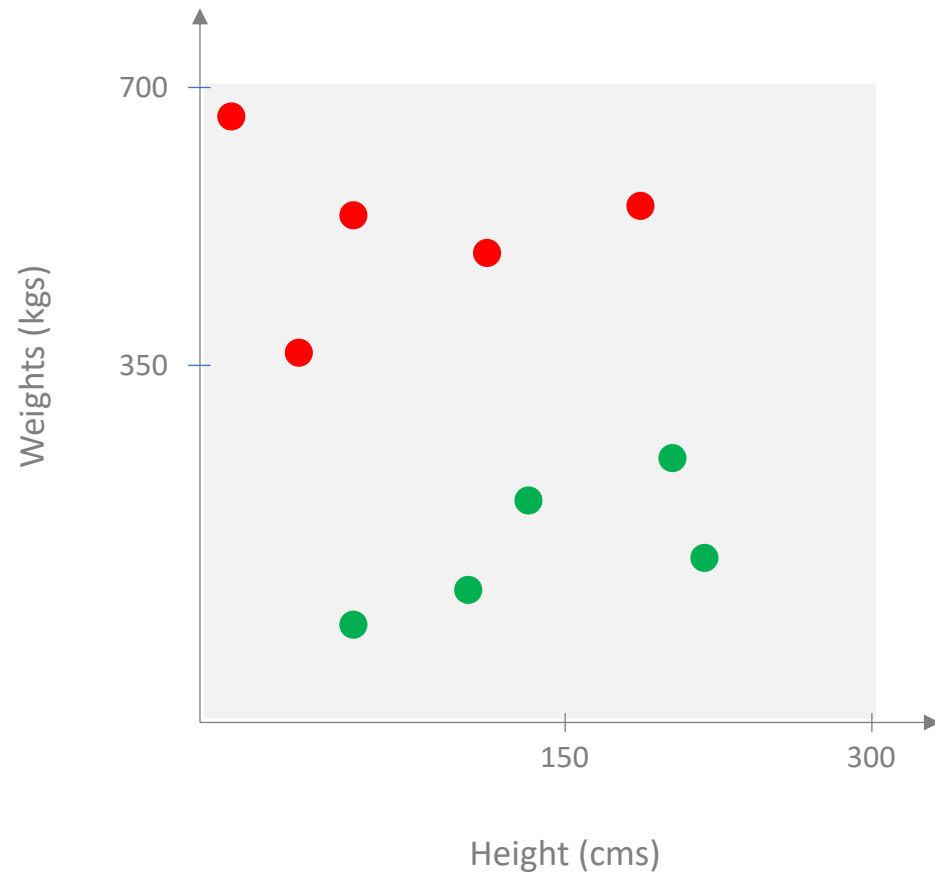




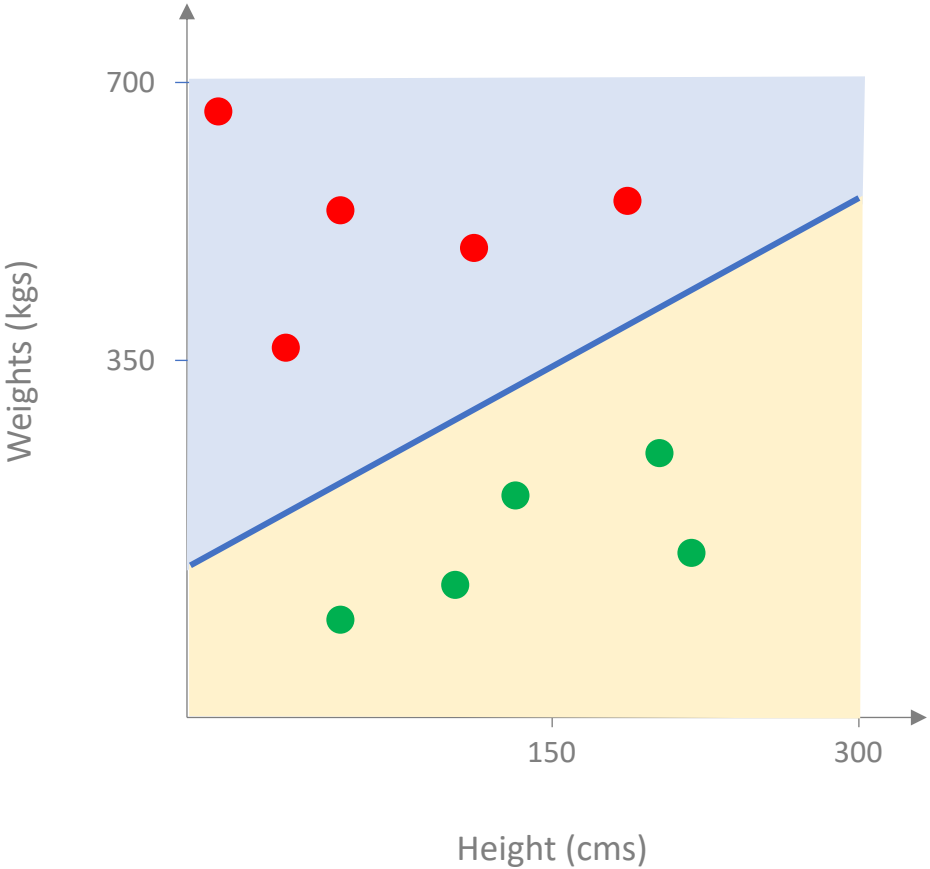
# The earlier linear model works!



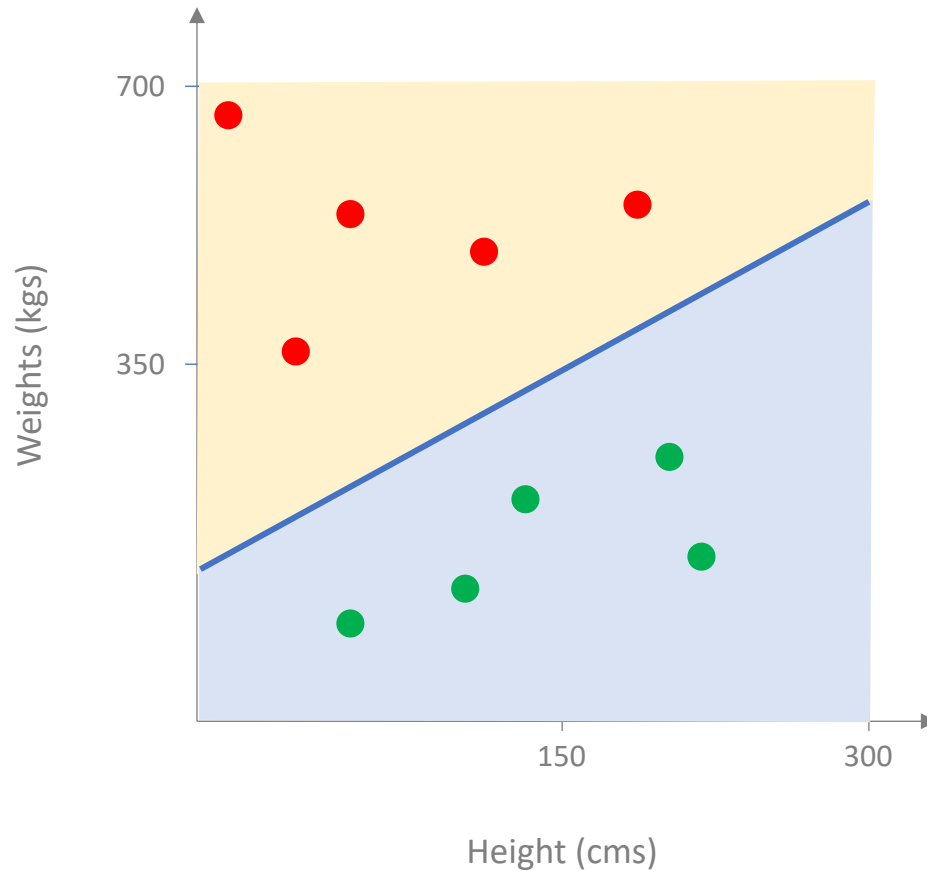
# What if the labels are flipped?



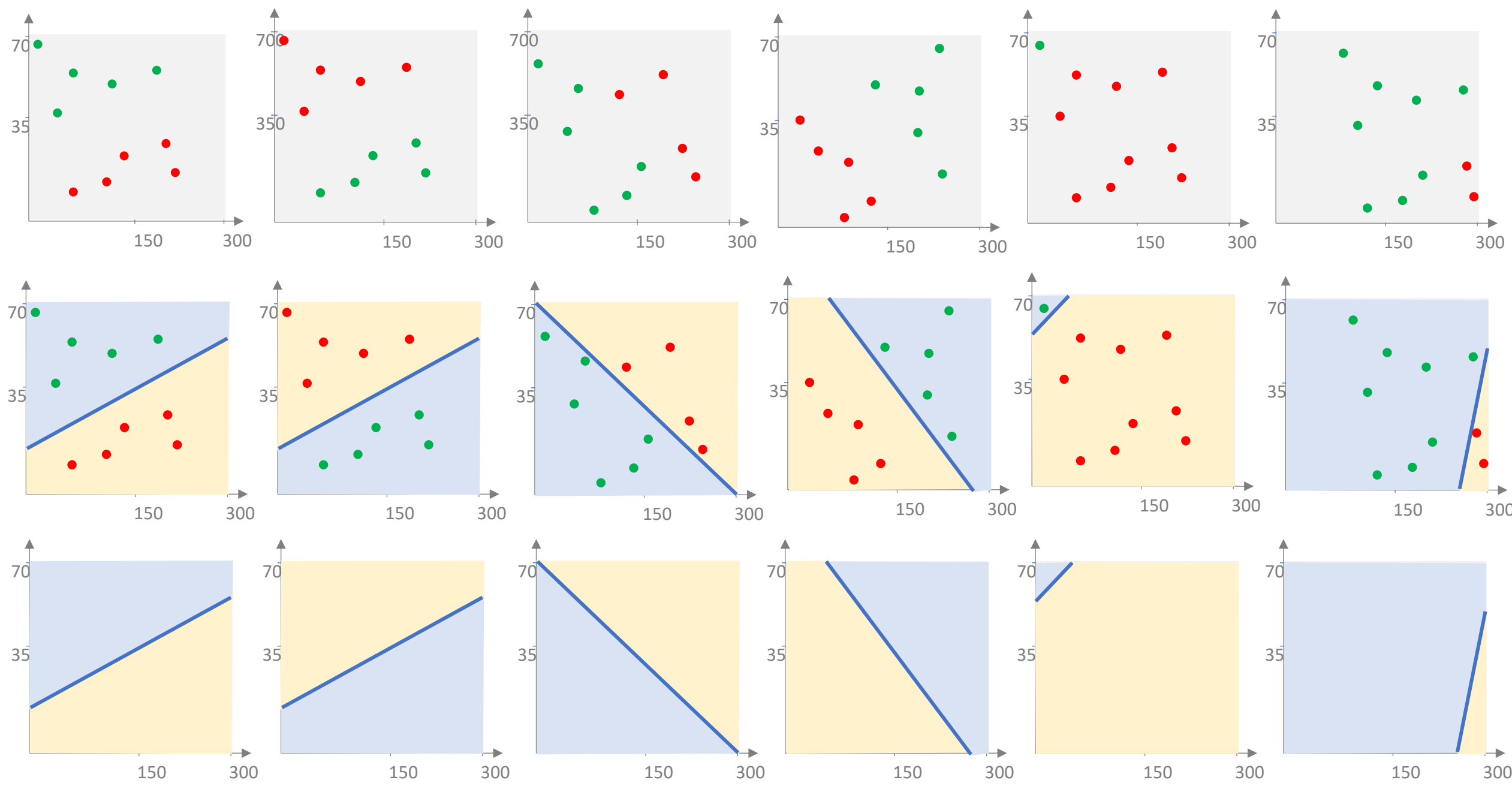
# The earlier linear model is terrible!



But the solution here is simple!



Is one linear model and its “complement” enough for all dataset?

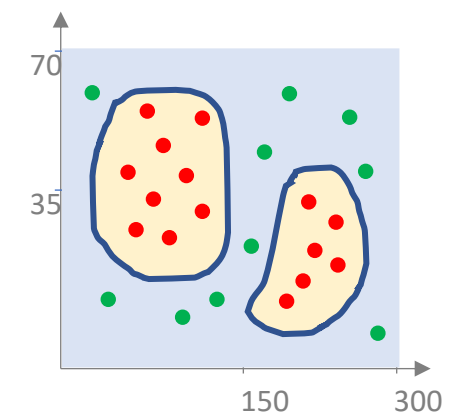
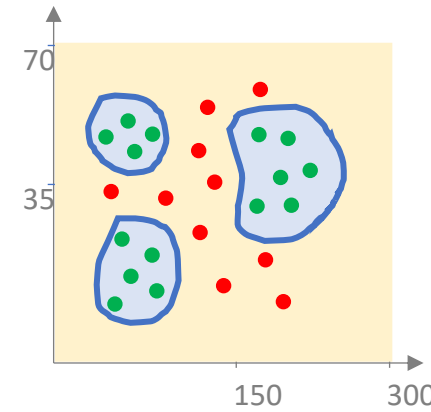
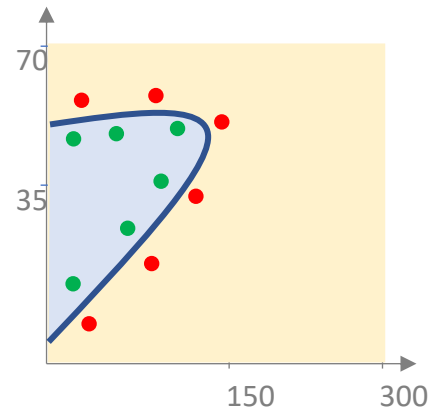
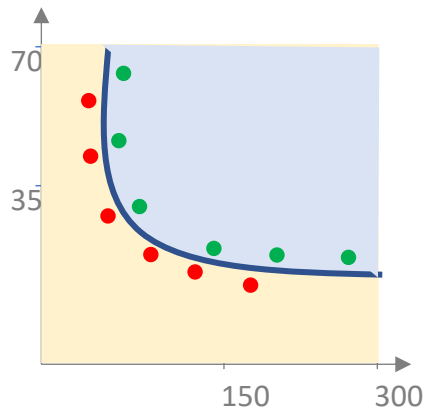
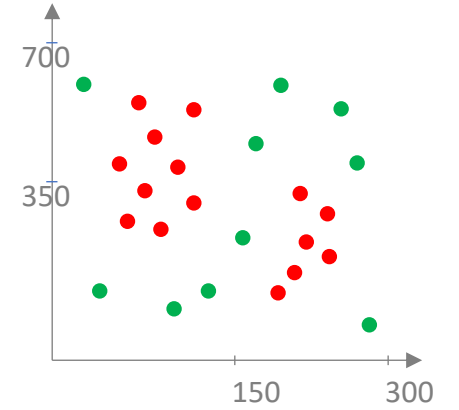
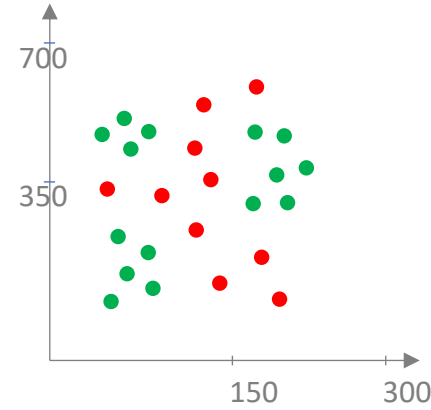
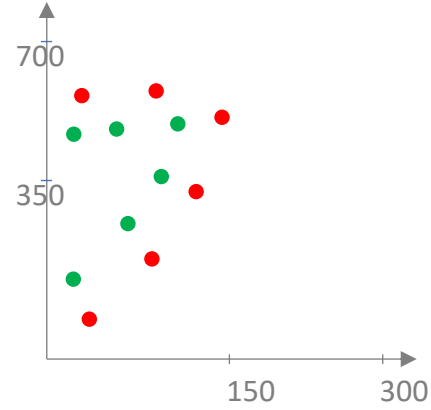
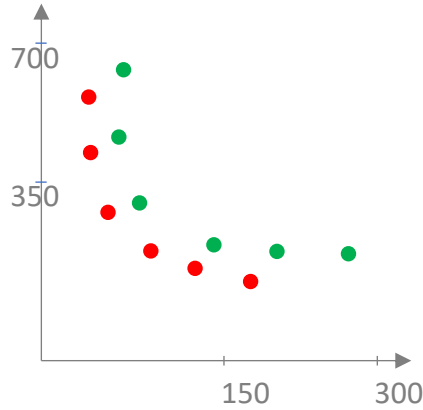


We need a *class* of models

Q1: Will linear models be enough for all datasets?

Q2: How should we define the *class* of linear models?

# Can a linear model explain any of these?

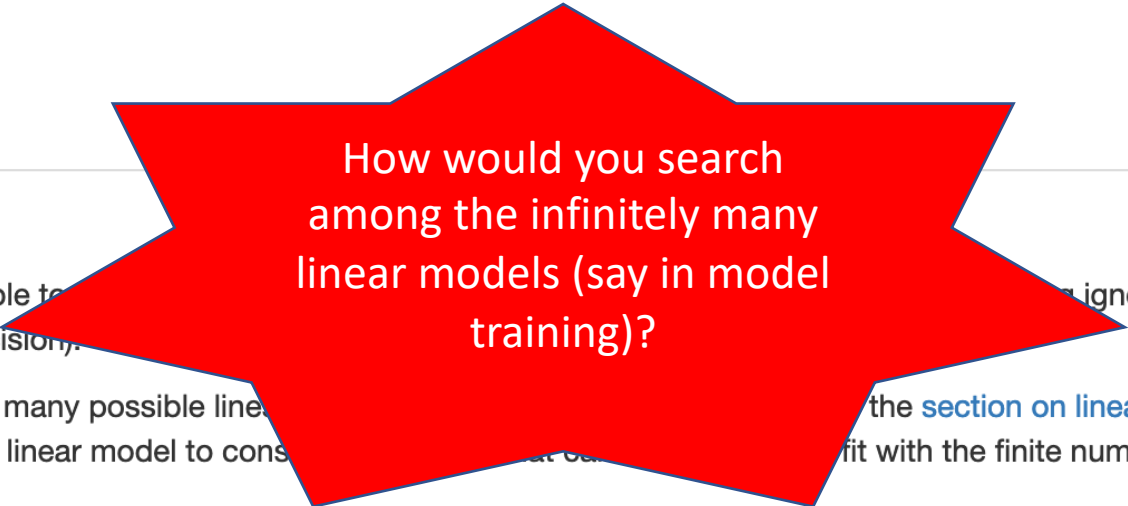


# Can a finite list of linear models work?

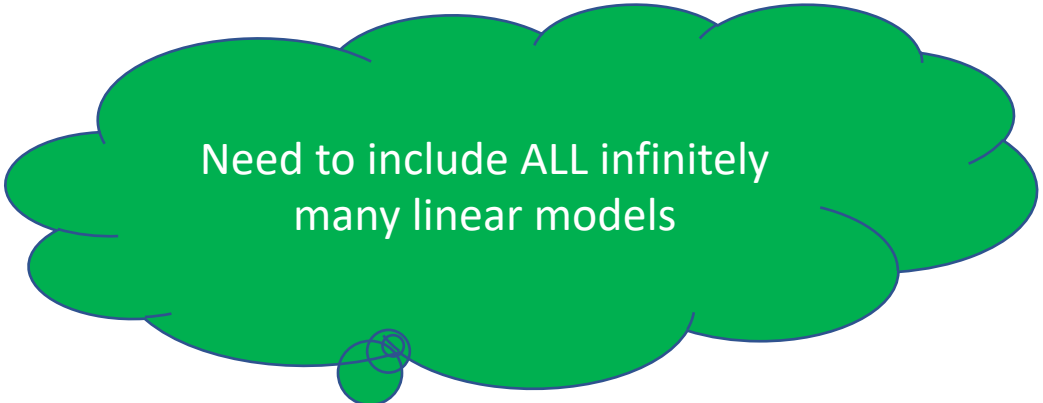
## Exercise

Convince yourself that having a *finite* set of linear models will not be able to represent a linear model without ignoring precision constraints: i.e. assume we can represent any number to arbitrary precision.

**Hint:** If you pick a finite set of linear models. Since there are infinitely many possible linear models (see the [section on linear models](#)). This means that your finite class will not represent a linear model. Can you use this linear model to construct a linear model that does not fit with the finite number of linear models in your class.



How would you search among the infinitely many linear models (say in model training)?



Need to include ALL infinitely many linear models



# Does there always exist non-linear model?

## Why Yes?

Convince yourself that given **any** dataset there is **always** a (possibly non-linear) model that fits it perfectly.

**Hint**: Given a dataset, can you use the dataset itself to define the model fits it perfectly? (Do not worry about how complicated the resulting model will be-- you just need to argue that such a model exists.) And do not peek below before you have spent some time thinking about the answer :-)

**NO: model training will not work**



# What properties should a model class have?

## Parsimonious representation

We would like to our model class such that its representation size does not depend on the size of your dataset (or has a more reasonable dependence than having to store the entire dataset to figure out the best fitting model).

In addition to representation size, models with small representation size can be considered to be the "right" model based on the [Occam's razor](#), which says that the simplest explanation is probably the best solution. This is also related to the notion of [generalization](#) (which we have [briefly seen before](#)).

It turns out that all the model classes that are used in an ML pipeline has this properties (though the notion of "small" changes from class to class).

## Efficient model training

In the later step in the ML pipeline of model training, the technical task is the following: given a class, figure out the best fit model from the given class. Of course, we would like to be able to do this step *efficiently*.

And it turns out that while sometimes it is possible to figure out the best fit model (e.g. [linear models](#)), this is not always possible (e.g. for [neural networks](#)). In the latter case, we often try to compute an approximation. If we find a model that has the smallest possible misclassification error, we might be able to find a model that makes

Are these properties enough?

## Efficient prediction

In the later step in the ML pipeline of prediction, the technical task is the following: given a specific model, figure out the label the model will assign to the point. Of course, we would like to be able to do this step *efficiently*.

We would like to point out that model training is basically done once while prediction is done over and over again once the training is done. Hence, while we want both efficient training and prediction, the notion of "efficient" is different for both tasks. E.g. in some cases, it is not unreasonable spend days trying to do model training but the corresponding prediction has to be done in (fraction of) a second.

# Other desirable properties

## Model expressivity

What is missing from the above properties is any constraint on how good the model class is at predicting datasets? In other words, for "real life" data, how accurately can this model class predict the correct labels?

The quotes around "real life" data is on purpose: it is a fool's errand to try and precisely define what real life data looks like. There are two ways around this: to propose certain natural conditions on the data and then theoretically show that under these theoretical conditions on the data, the best model in your class will predict the data labels with small error (and presumably you also experimentally show that at least some representative datasets satisfy the theoretical conditions). Alternatively, you run experiments and show that on some representative datasets the *empirical* prediction errors of the best model in your class is small.

## Other desirable properties?

As we progress in the course, we will see that there are properties other than the ones above (which are the primary focus of traditional ML) that could be useful. E.g. does the model make it easy to "explain" its decision to a human? Can we argue that in some precise sense the model is "fair" or not "biased."

Can you think of other useful properties?

Break!

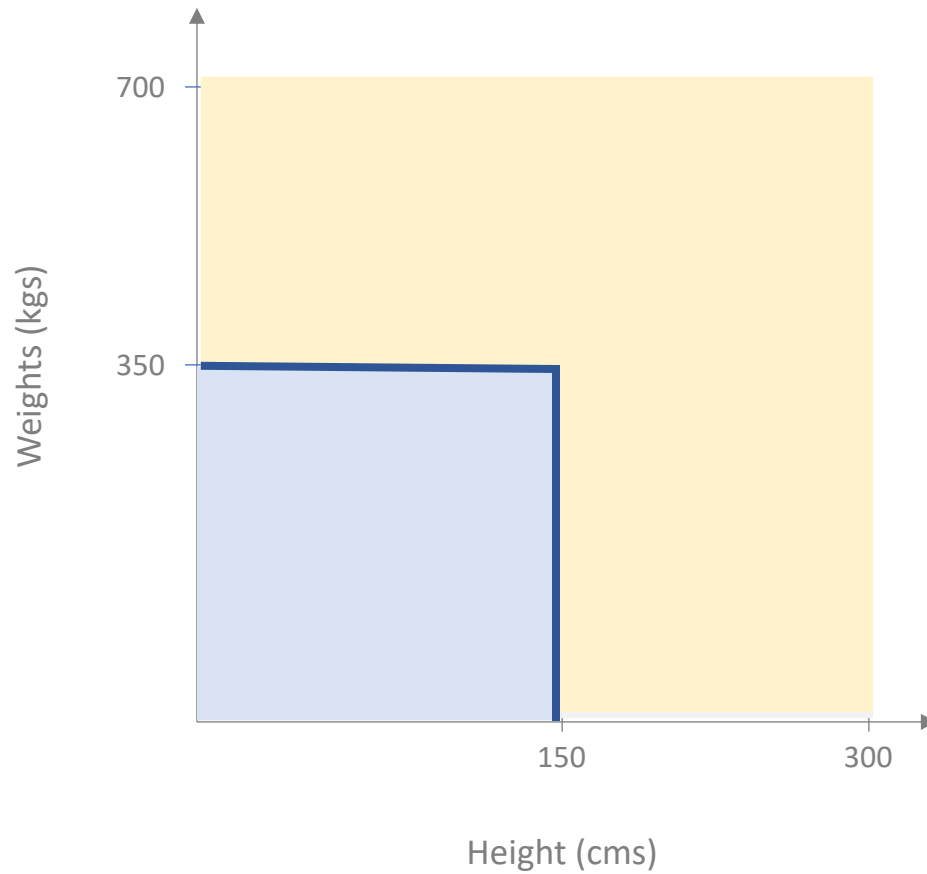
# Decision tree models

Heard of 20 Questions?





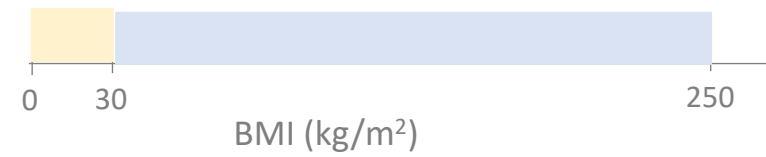
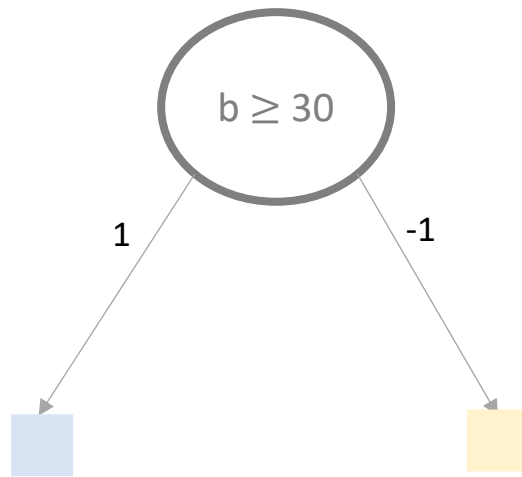
# We already have seen a decision tree



$$f(w, h) = \begin{cases} 1 & \text{if } w < w^* \text{ and } h < h^* \\ -1 & \text{otherwise} \end{cases}$$



# Decision trees in one variable



# Asking more questions



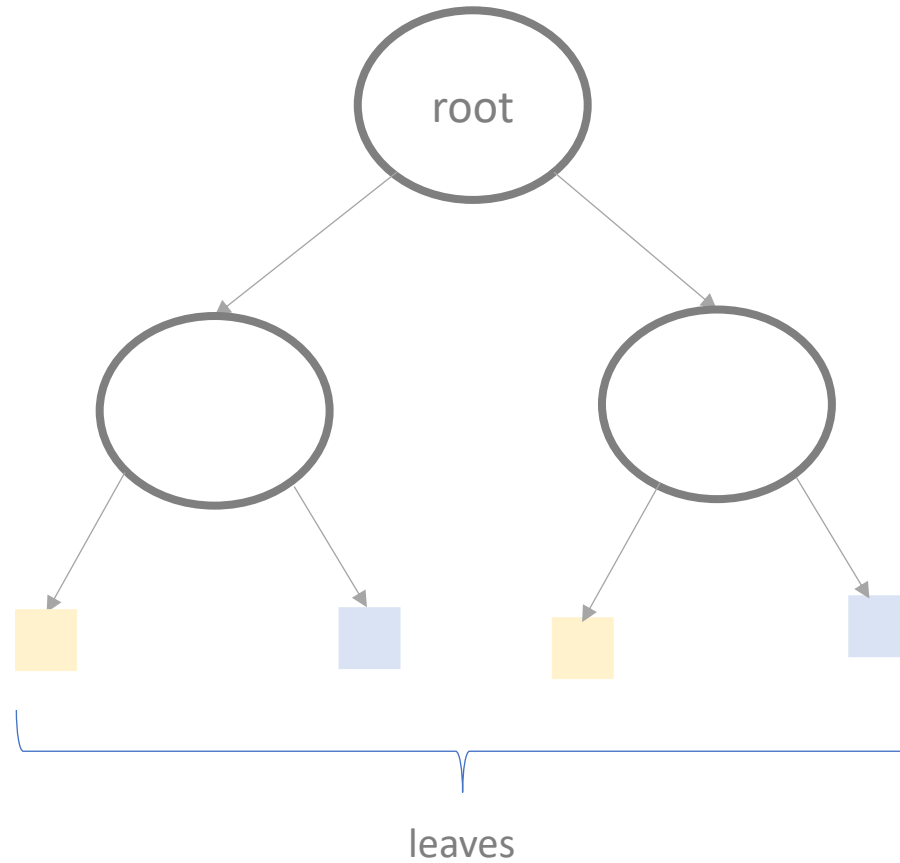
## Class of decision trees in one variable (formal-ish definition)

A decision tree is formally defined as a **labeled binary tree** (i.e. you have nodes, other than leaf nodes, with two outgoing labels-- one labeled as 1 and the other as  $-1$  and each leaf is colored blue or yellow. Note that a leaf node has no outgoing edge). Further each node other than the leaf nodes (also called **internal nodes**) have a comparison of the form  $b \geq b^*$  associated with them).

The class of decision tree models is the collection of all possible decision trees in one variable. Note that this is indeed an infinite class like the class of linear models but as we will see shortly, this class is way more expressive.



# Why is this a tree?





# Last model class: Neural networks (NNs)

Home / Machine learning & AI



29,317 views | Feb 9, 2020, 08:39pm

FEBRUARY 21, 2020

## Deep learning AI discovers surprising new antibiotics

by Sriram Chandrasekaran, The Conversation



AI



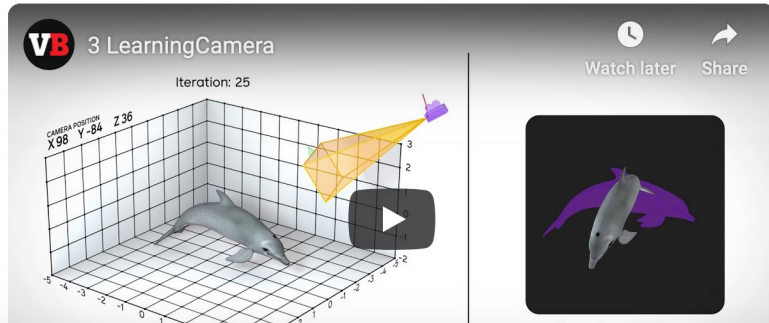
Facebook launches 3D deep learning library for



PyTorch



KHARI JOHNSON @KHARIJOHNSON FEBRUARY 6, 2020 9:00 AM



### VB TRANSFORM

The AI event for business leaders

San Francisco  
July 15 - 16

A colored electron microscope image of MRSA. Credit: NIH - NIAID/fli...

Imagine you're a fossil hunter. You spend months in the heat of Arizona

## Deep Learning Has Limits. But Its Commercial Impact Has Just Begun.

### Deep learning godfathers Bengio, Hinton, and LeCun say the field can fix its flaws

Yoshua Bengio, Geoffrey Hinton, and Yann LeCun took the stage in Manhattan at an AI conference to present a united front about how deep learning can move past obstacles like adversarial examples and maybe even gain common sense.

By Tiernan Ray | February 10, 2020 -- 14:02 GMT (06:02 PST) | Topic: Artificial Intelligence



From left, Geoffrey Hinton, Yann LeCun, Yoshua Bengio.

Studies have shown that AI can outperform human doctors at identifying breast cancer from ... [+]

UNIVERSAL IMAGES GROUP VIA GETTY IMAGES

From IPsoft

IPsoft Inc. Open

DIGITAL Workforce.ai  
Digital Employees for Hire  
Hire a Digital Employee

MORE FROM TIERNAN RAY

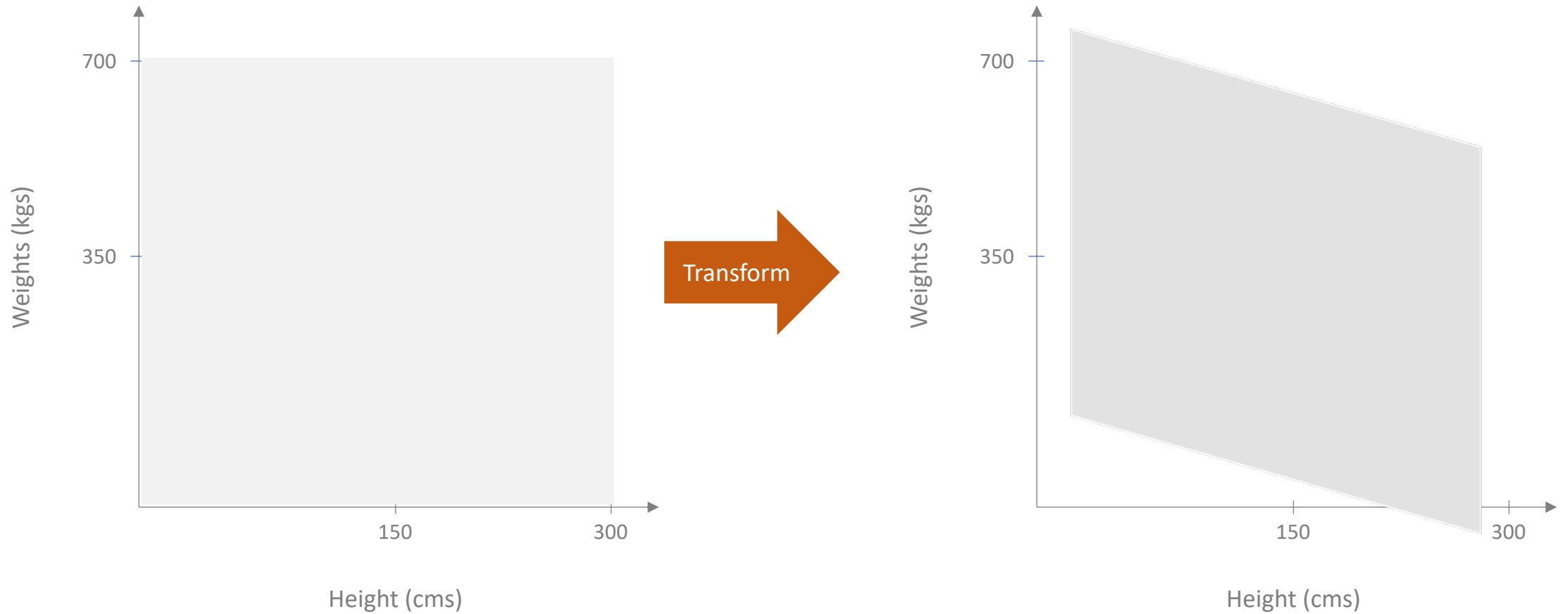
Artificial Intelligence  
Google AI chief Dean sees evolution in ML Perf benchmark for

# Away from the hype: non-deep NNs

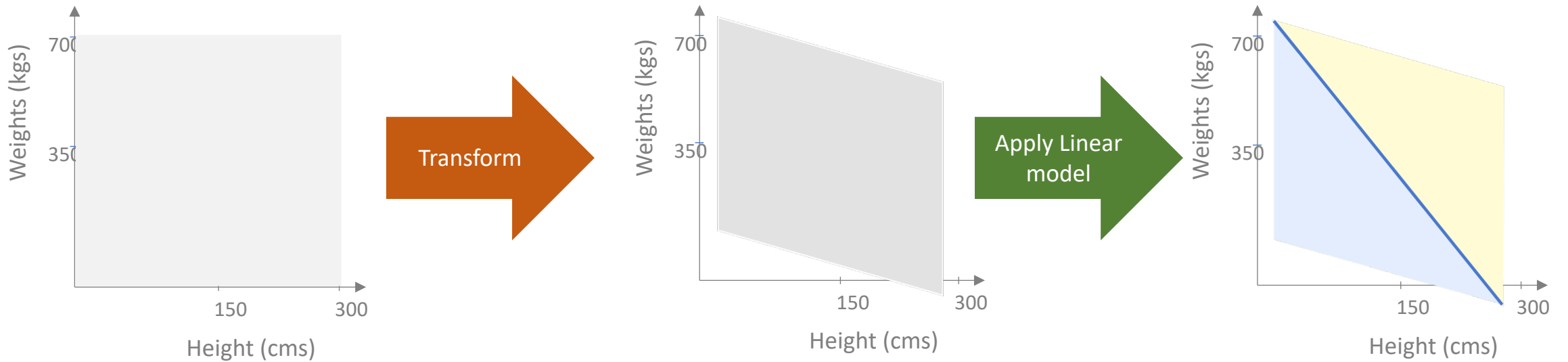
## Simplification galore!

Much more so than the other two models, we will be simplifying the treatment of neural networks. In particular, we will be presenting a very strict sub-class of neural networks. Later on, we will briefly mention the generalization needed to do [deep learning](#), which is a fancy re-branding of ML techniques that have existed for decades and folks are [already wondering if deep learning is over-hyped](#). But that is a story for another time.

# NN Idea 1: Transform the underlying space



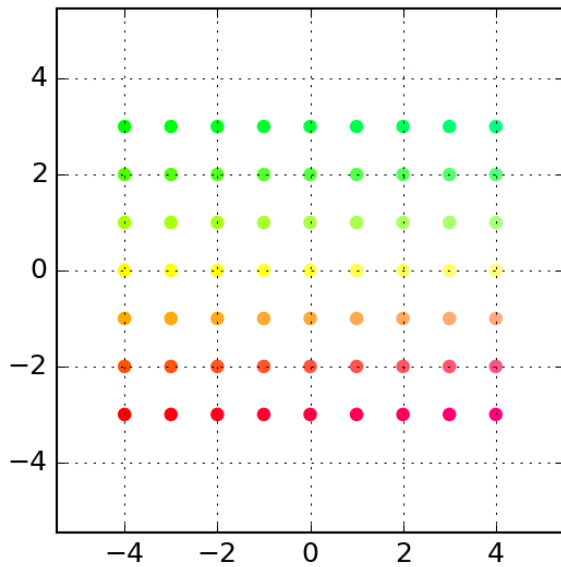
# NN Idea 2: Apply linear model AFTER transform



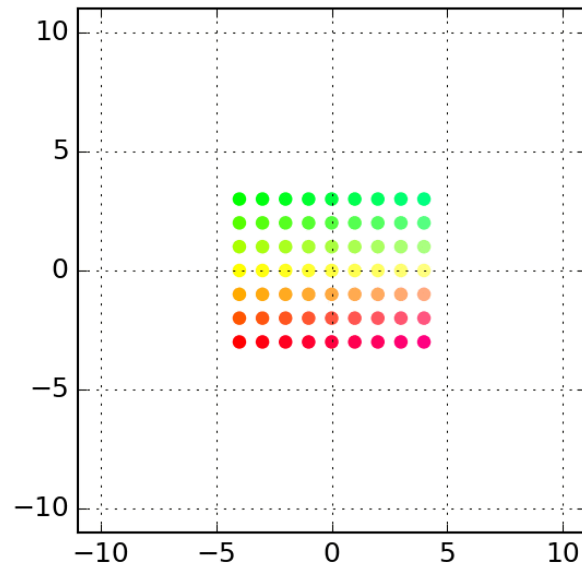


# Whither the transforms?

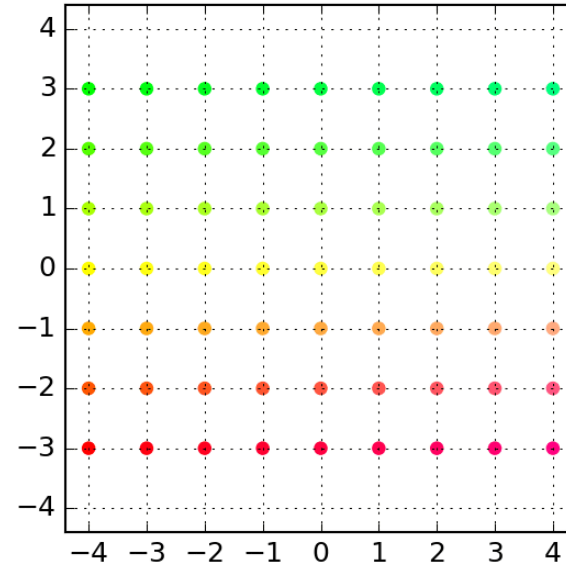
One possibility: Linear transforms



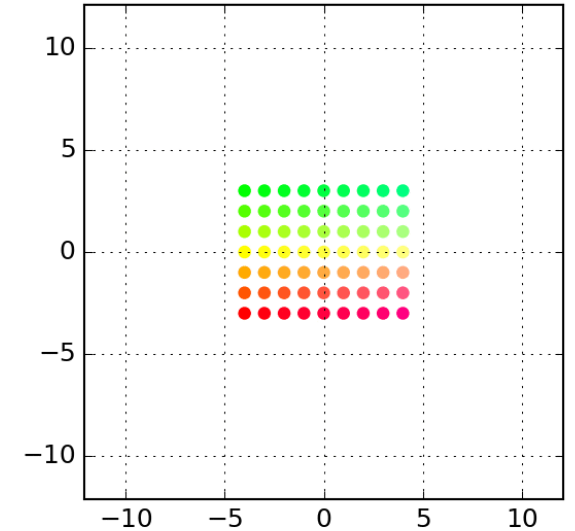
Rotation



Shear

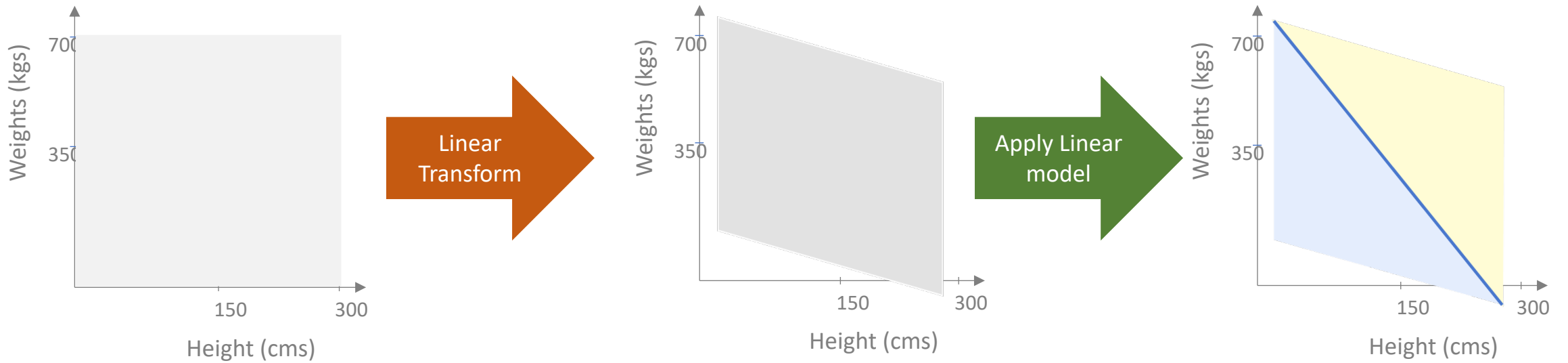


Permutation

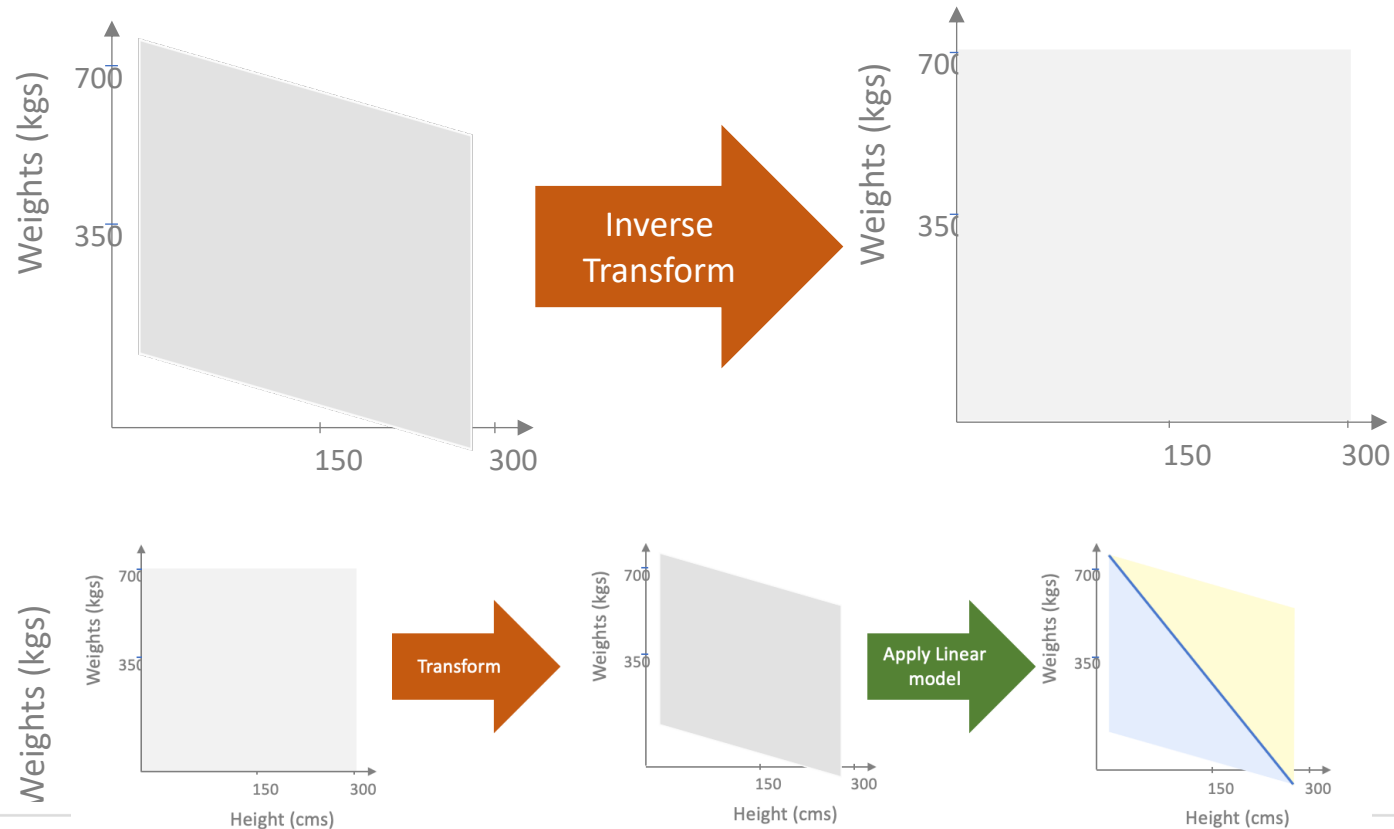


General Linear

# Current overall scheme



# Linear transforms\* are invertible



## Exercise

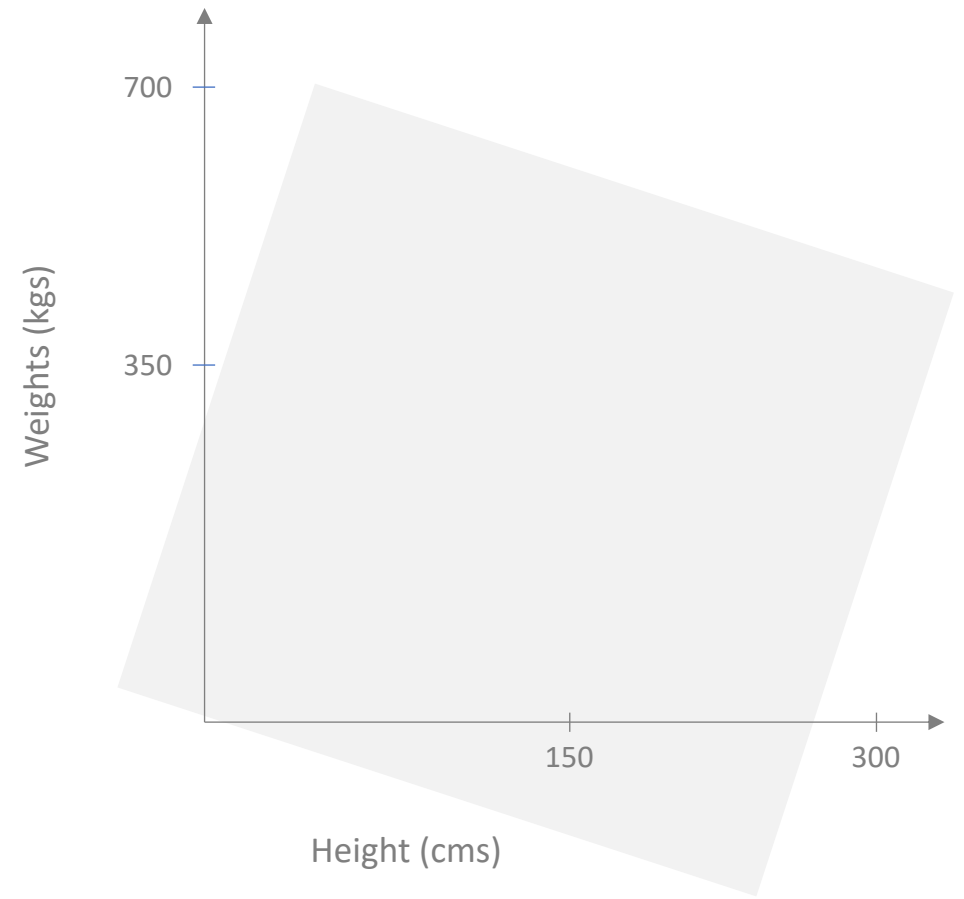
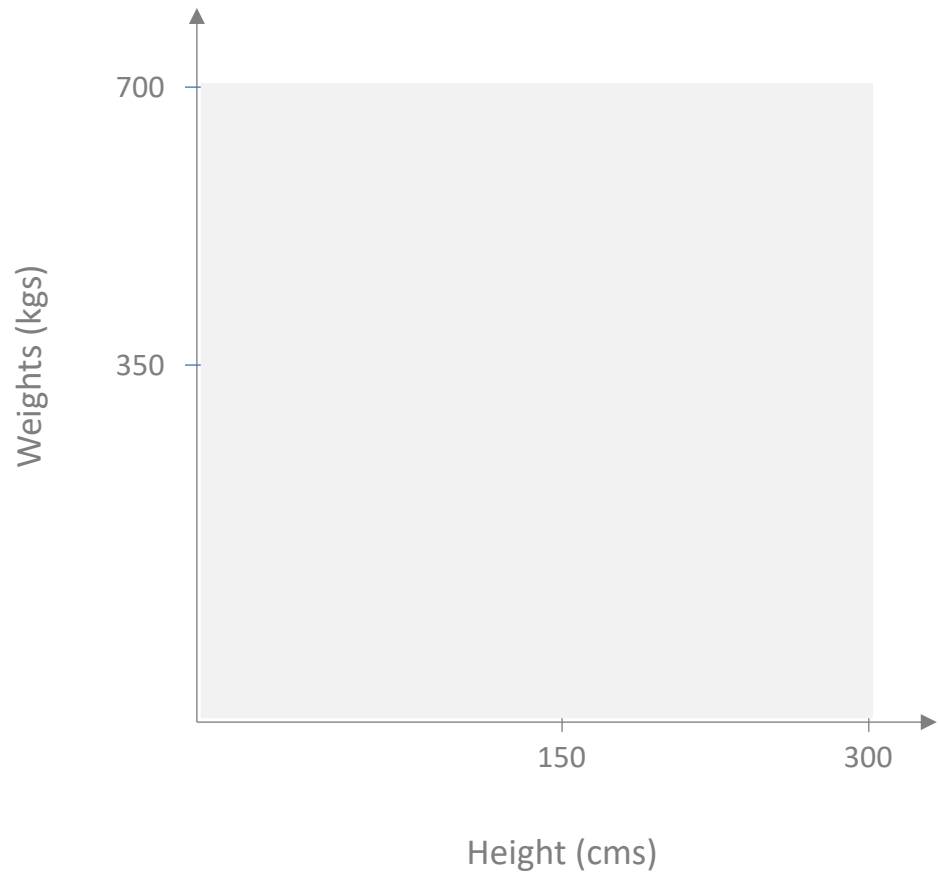
Argue that a linear transform followed by a linear model is **equivalent** to (another) linear model in the original space.

**Hint:** Can you "reverse" the whole process?

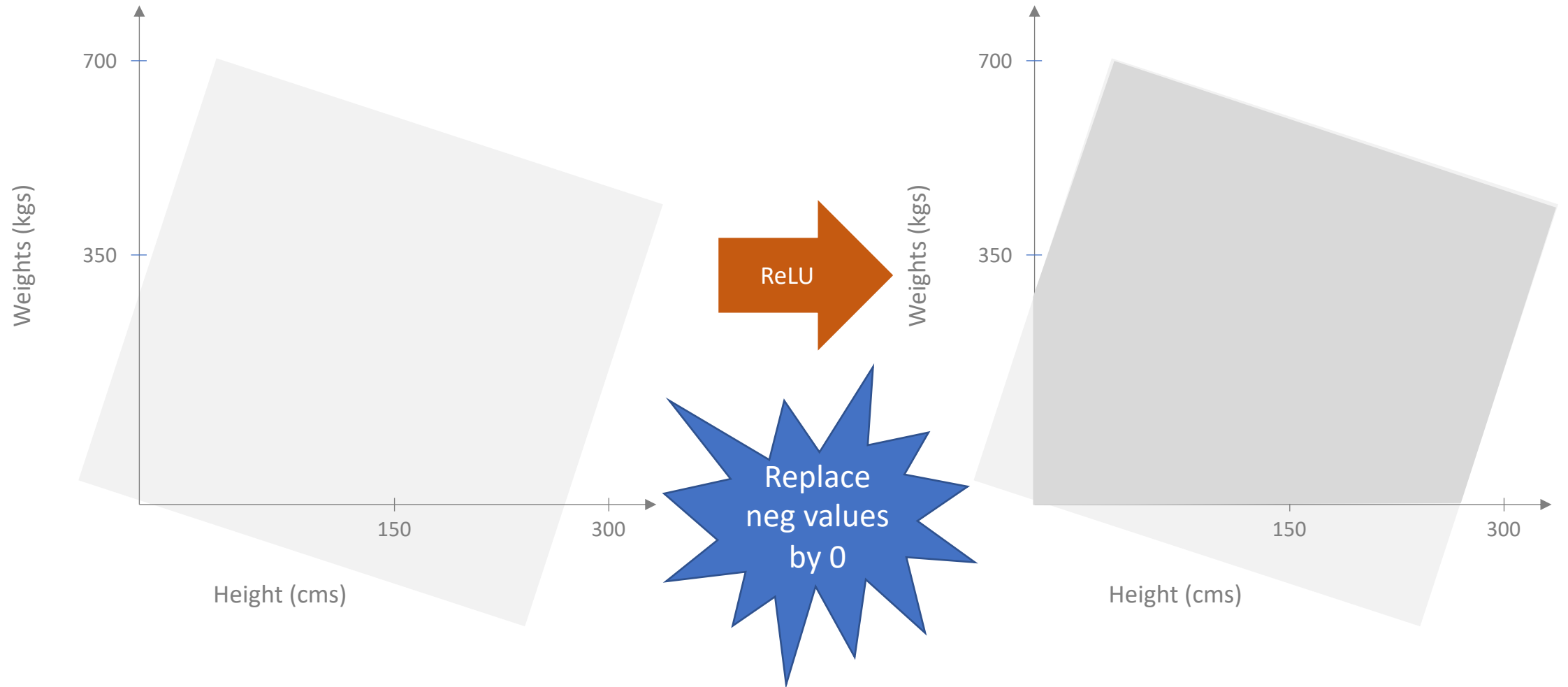
Height (cms)



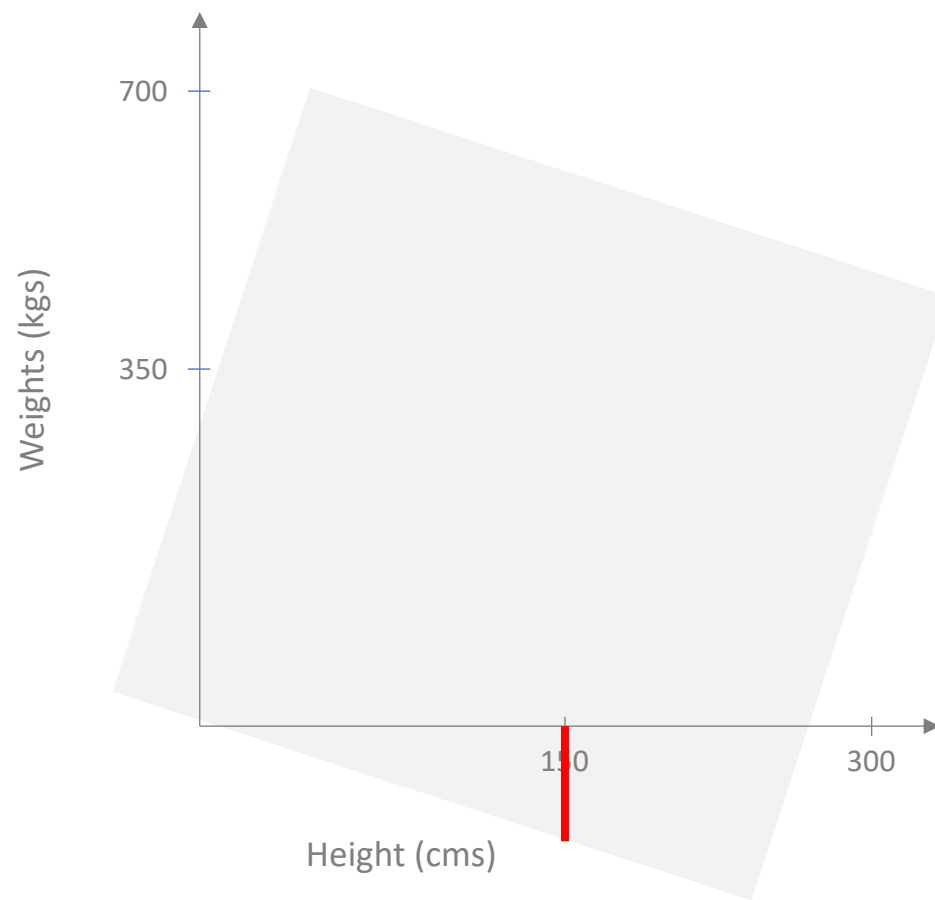
# NN Idea 1': Use a non-linear transform



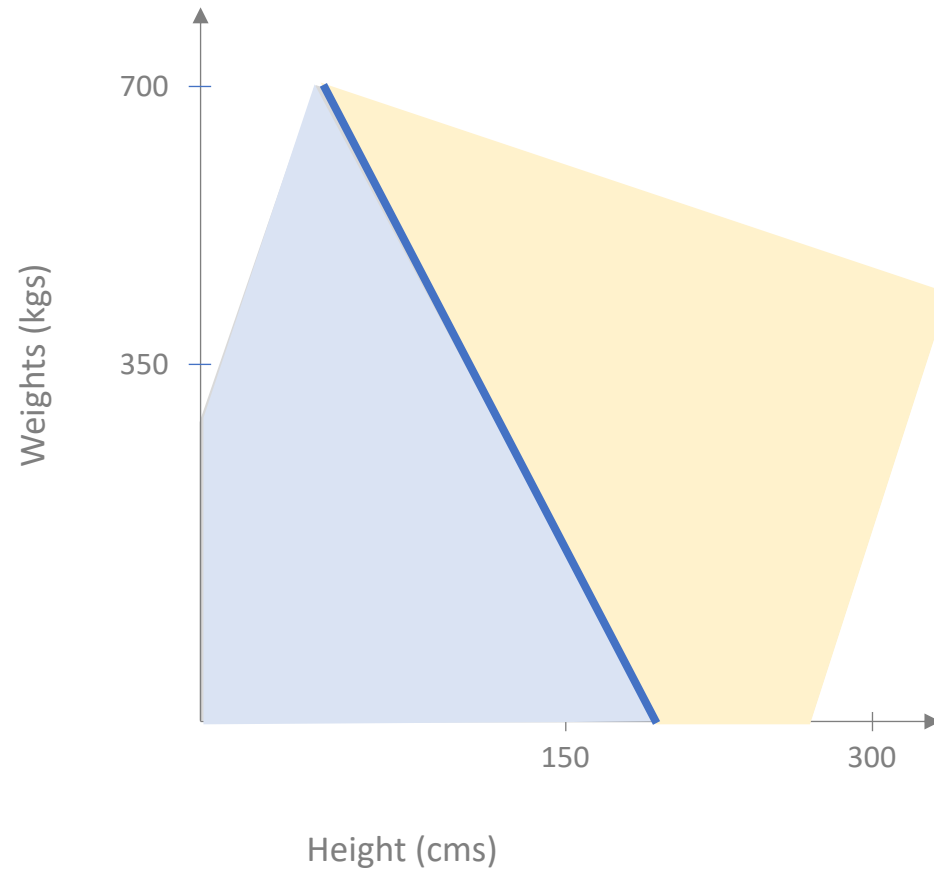
# NN Idea 1': Use a non-linear transform



# Bunch of values get zeroed out!

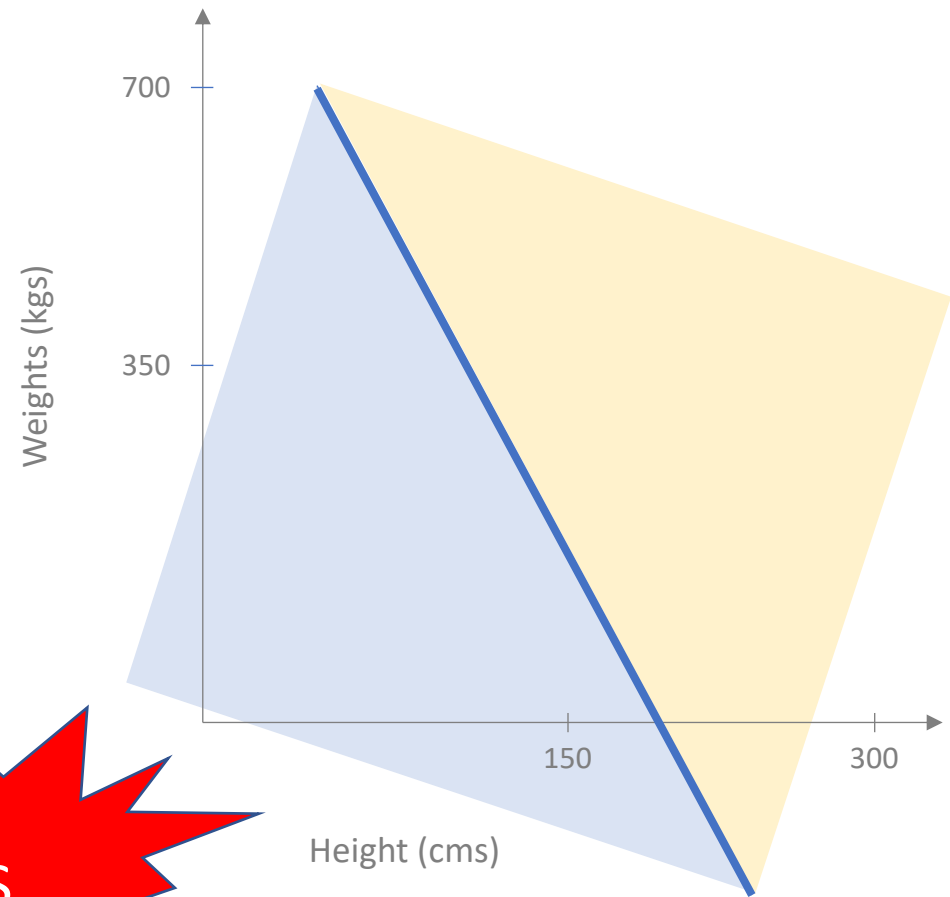
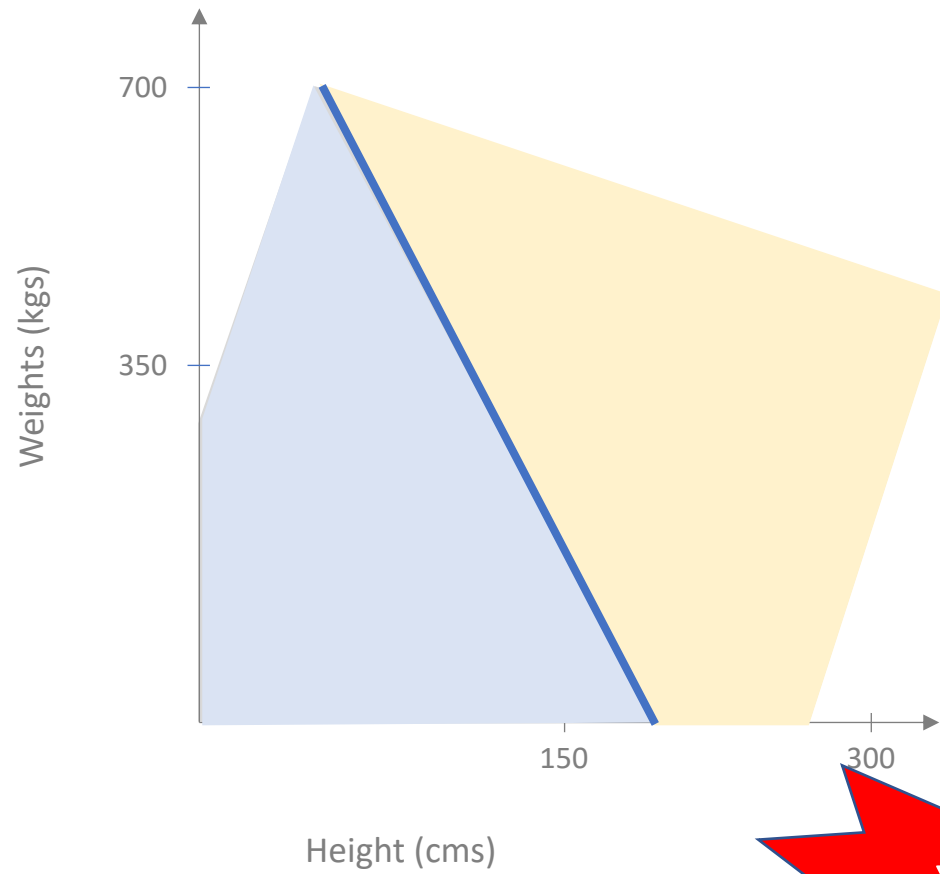


# Let's apply linear model to transformed space



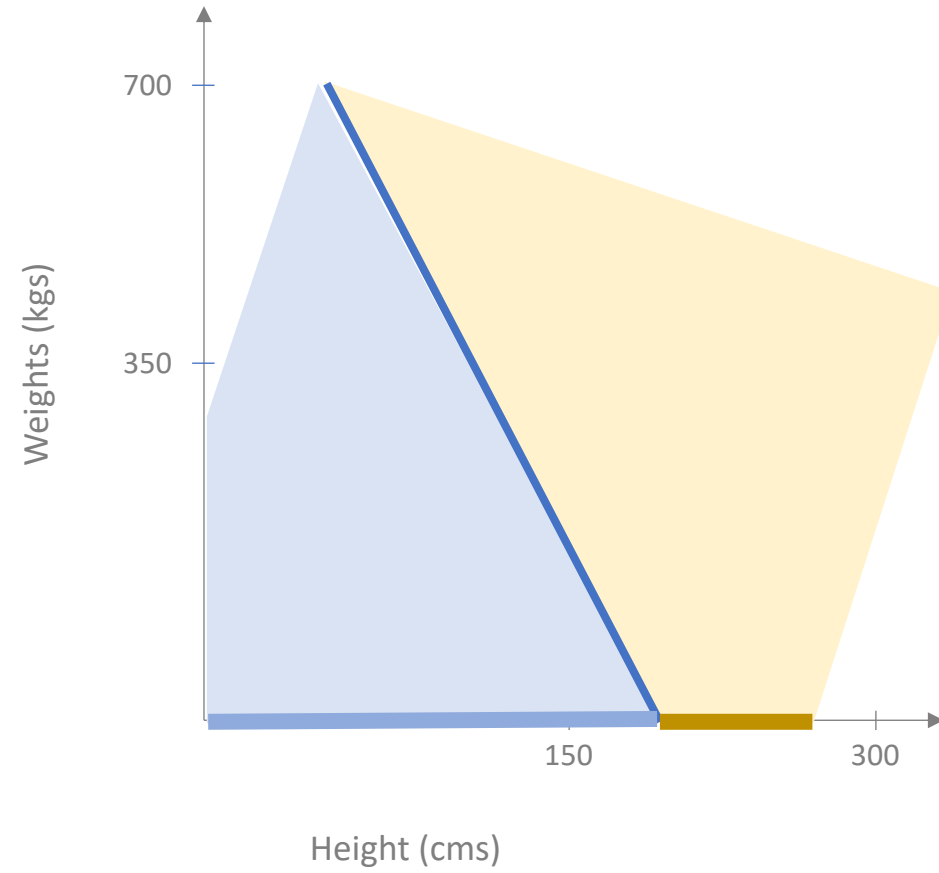


# Did we gain over linear models?

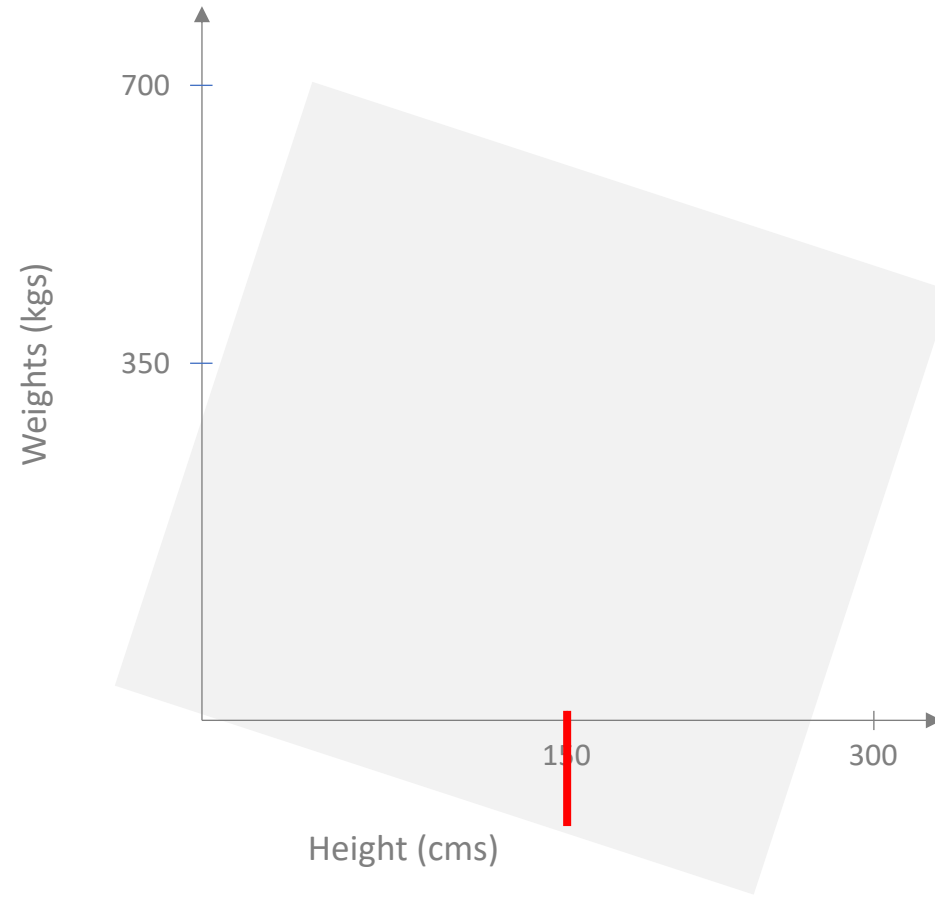


YES

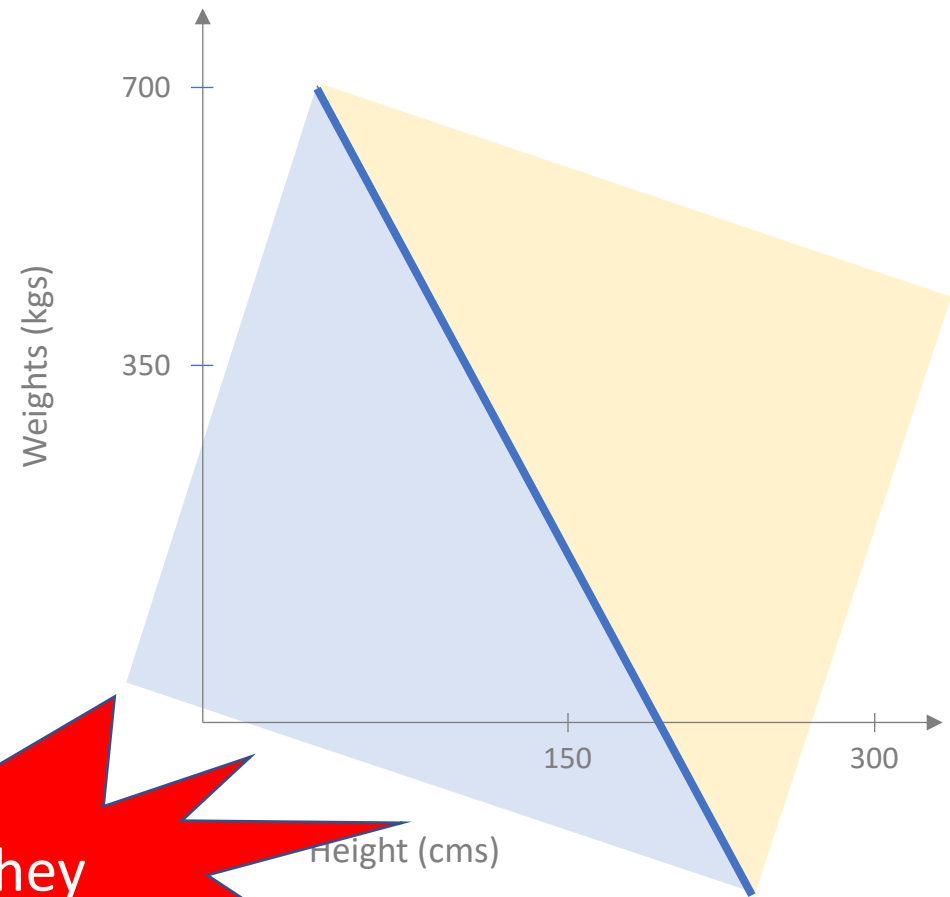
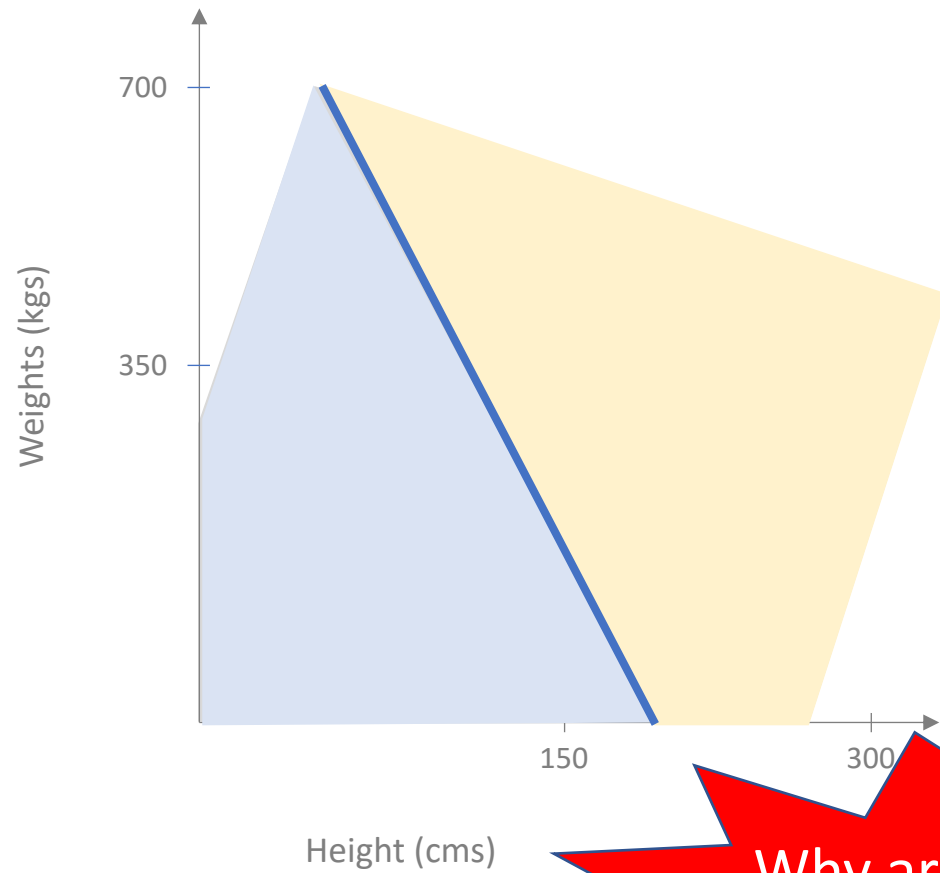
# Reminder # 1



# Reminder # 2

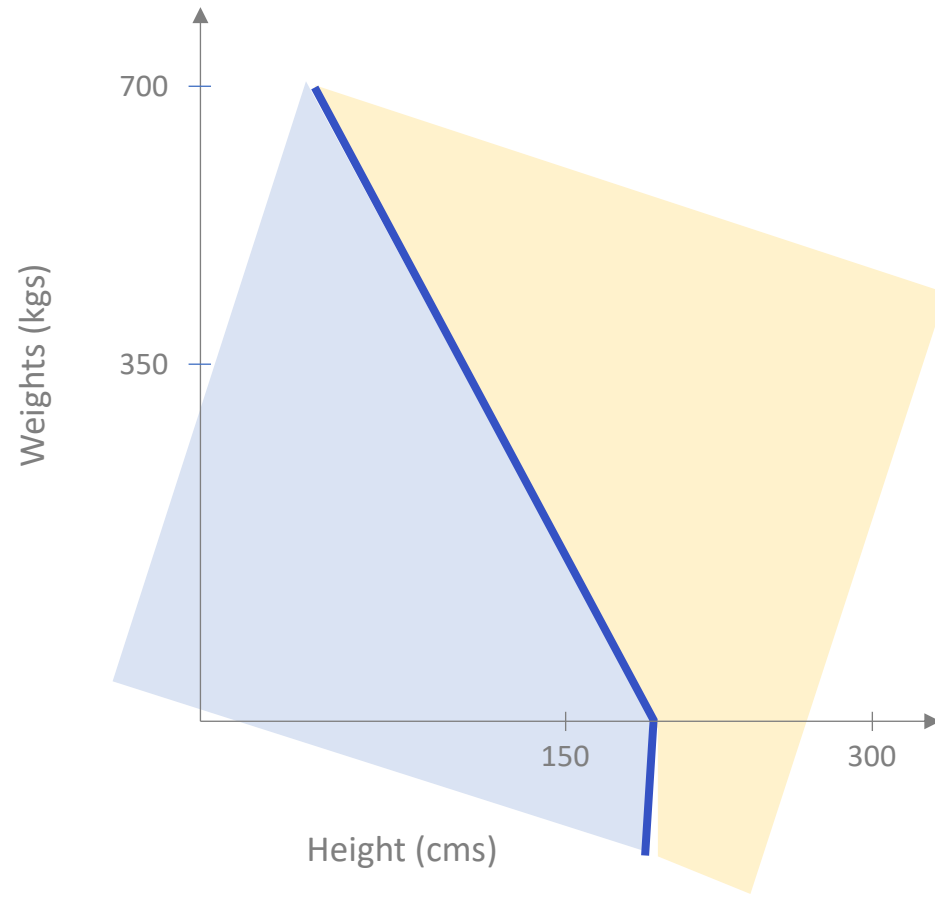
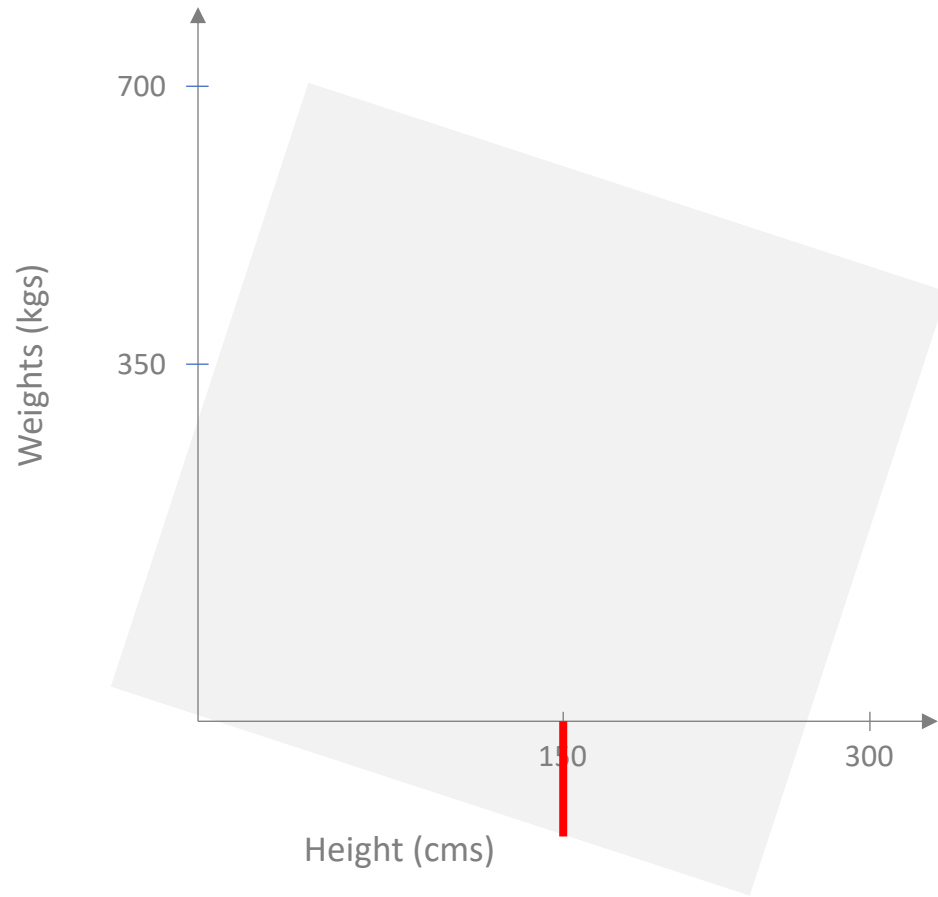


# Did we gain over linear models?

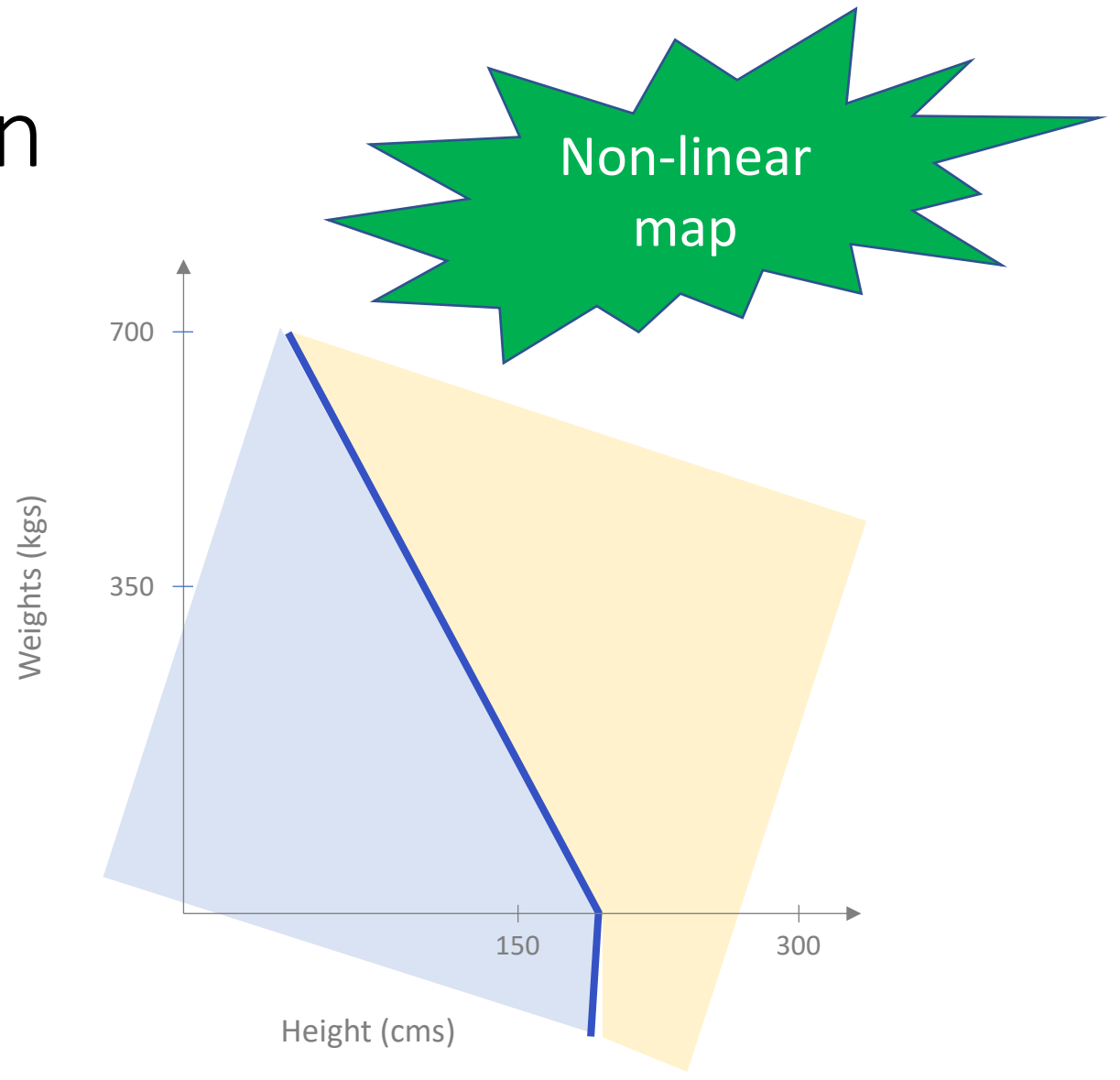
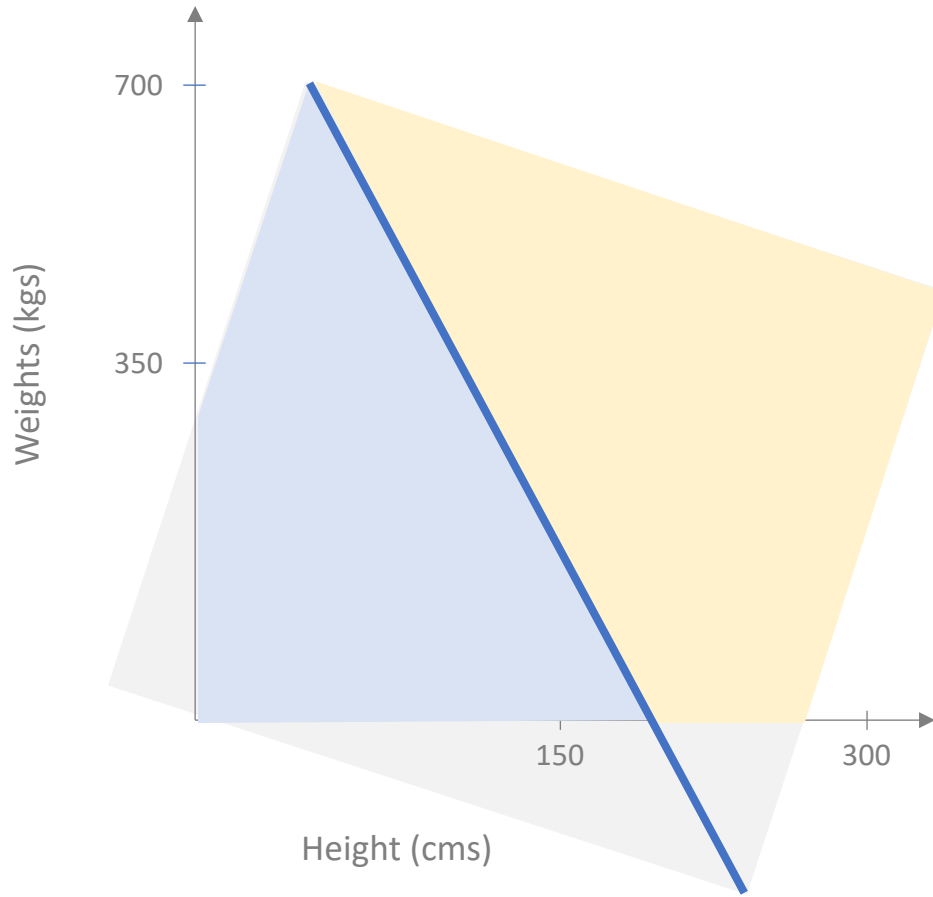


Why are they  
different?

# What is really happening....



# Side by side comparison





# An exercise to do at home (if you so choose)

## Exercise (Neural networks in one variable)

We now briefly give a strong reason for why we did not consider neural networks in one variable-- it turns out that this class of models is **exactly** the same as the class of linear models in one variable. In this exercise, you will argue this claim.

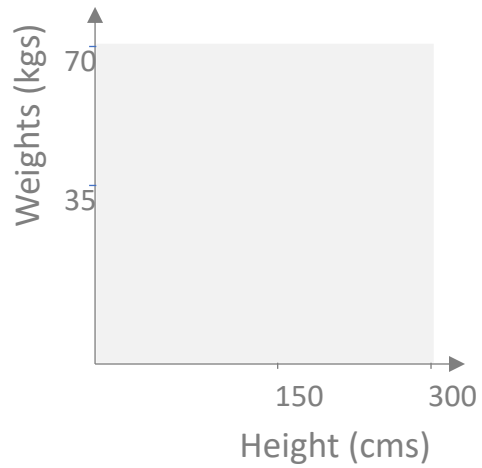
First we more precisely define neural networks in one variable. First a linear transform in one variable is a map  $b \mapsto m \cdot b + c$ . So if  $\ell$  is the final linear model that we will apply after applying ReLU to the linear transform of  $b$ , then we get the following description of a neural networks in one variable:

$$NN(b) := \ell(\text{ReLU}(m \cdot b + c)).$$

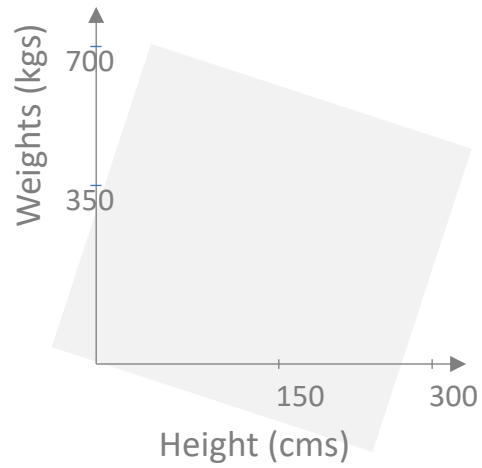
Argue that for any linear model  $\ell$ , the model  $NN$  is also a linear model.



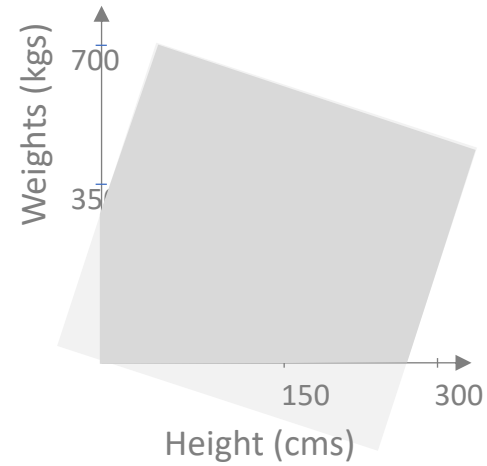
# What is so “deep” about this?



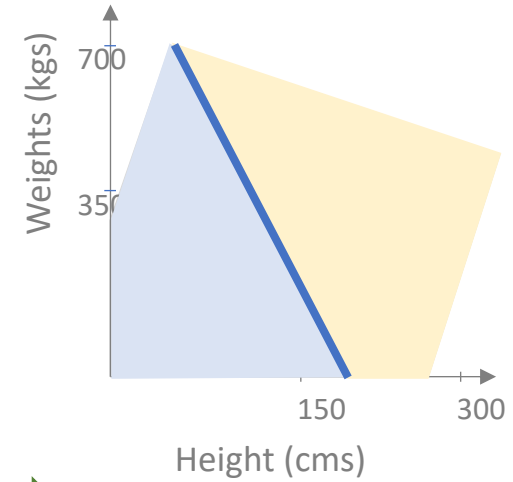
Linear Transform



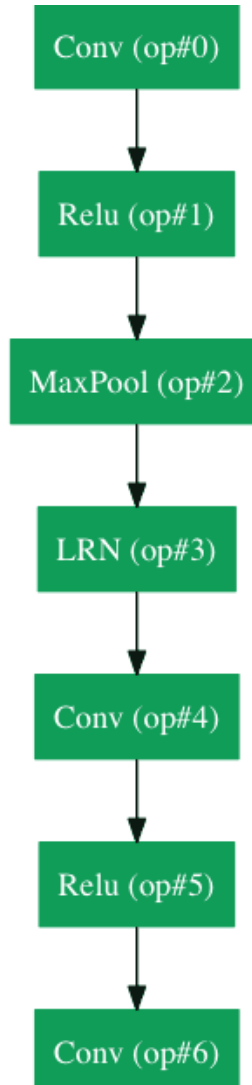
ReLU



Apply Linear model



# Use many non-linear transforms!



# Desired properties of NNs

## Model expressivity

It is [known that neural networks can approximate any functions](#) and hence can represent any labeled dataset exactly. Covering this result is out of the scope of this course.

## Parsimonious representation

It turns out that neural networks (at least the recent ones with multiple layers) actually are **not** parsimonious. In fact, these models are **overparameterized**. Given this, it is somewhat surprising that [in many widely used applications neural networks do not seem to overfit](#). Covering this result is out of the scope of this course.

## Efficient model training

Not much is known theoretically on the efficiency of model training for neural networks. In practice, algorithms such as [\(stochastic\) gradient descent](#) work well in practice but its theoretical properties are not well understood in the context of training neural networks. We will very briefly come back to gradient descent when we consider the model training steps of the ML pipeline.

## Efficient prediction

The efficiency prediction depends on the how many layers are used in the neural network as well as the efficiency of computing each layer. Covering this out of the scope of the course.

## Other desirable properties?

Neural networks are notorious for **not** being explainable. We will return to this topic later in the course.