Machine Learning

Lecture 3 – Generalisation, Training & Test Set, Representation

Dr SHI Lei

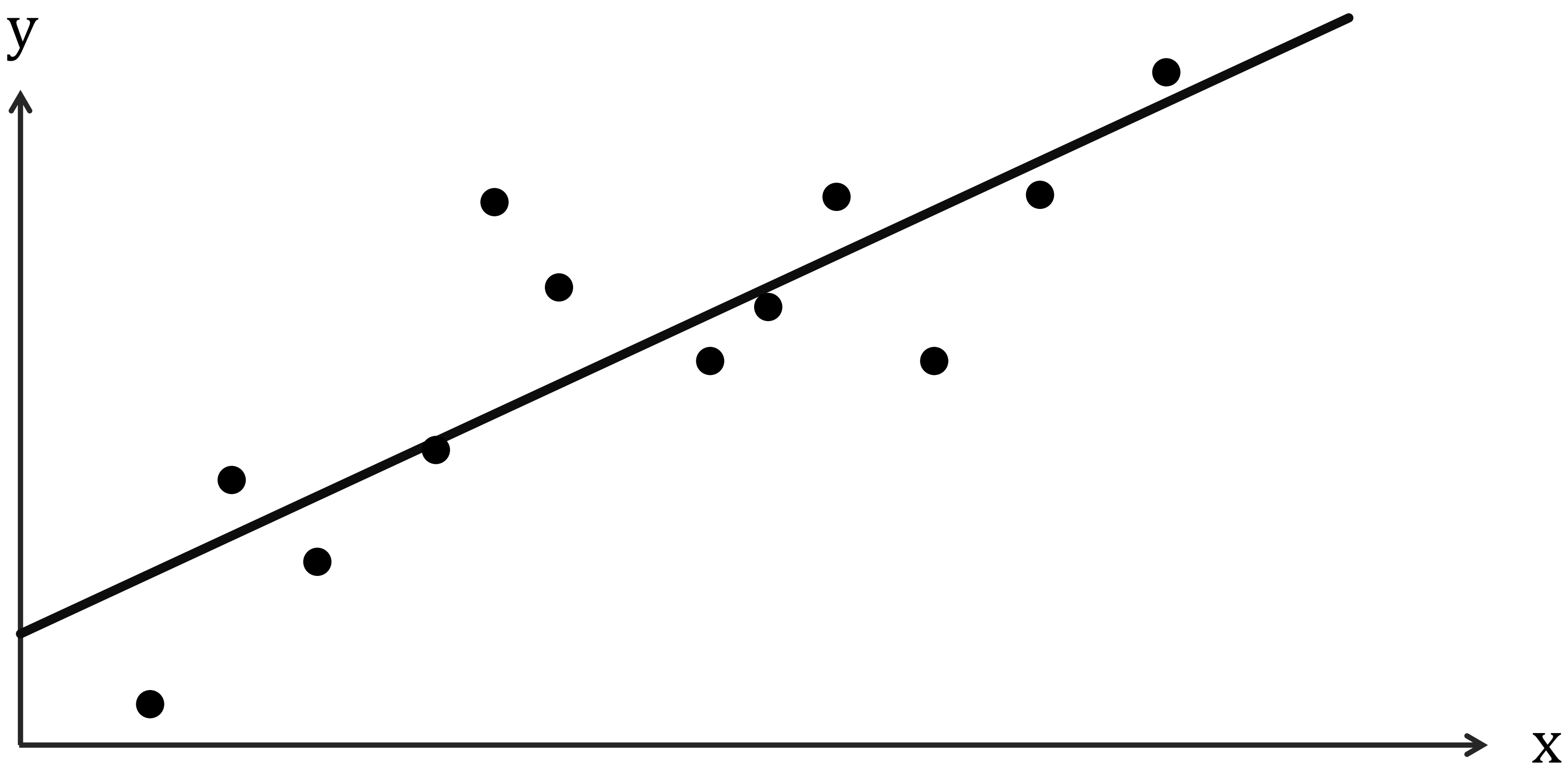
Last Lecture

- Linear Regression
- Training & Loss

Last Lecture

Linear Regression

A method to find the straight line or hyperplane that best fits a set of points.



$$y = w\,x + b$$

Last Lecture

Linear Regression

$$y = b + wx \qquad \Big| \qquad \hat{y} = b + w_1 x_1$$

$\hat{y}$ the predicted label (a desired output).

b the bias (the y-intercept), sometimes referred to as w_0.

$\mathbf{w_1}$ the weight of feature 1 (slope).

$\mathbf{x_1}$ a feature (a known input).

To infer (predict) the exam mark $\hat{y}$ for a new coursework mark value x_1, just substitute the x_1 value into this model.

Last Lecture

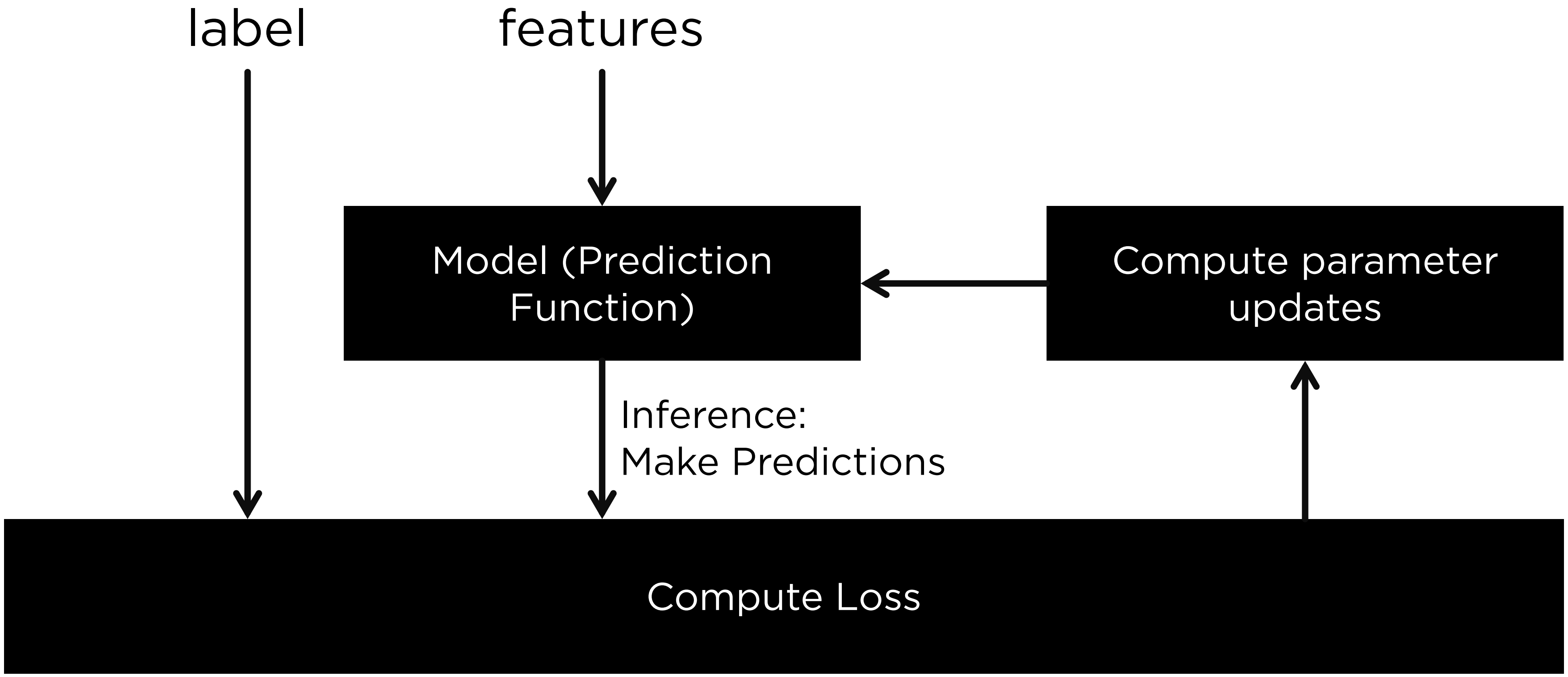
Training & Loss

- **Training** a model: learning (determining) good values for all weights and the bias from labelled examples.
- **Loss**: the penalty for a bad prediction. $MSE = \frac{1}{N} \sum_{(x,y) \in D} (y - prediction(x))^2$
- **Empirical Risk Minimization**: the process of examining many examples and attempting to find a model that minimise loss.
- **Goal of training**: to find a set of weights and biases that have <u>low loss</u>, on average, across all examples.

Last Lecture

Gradient Descent

Repeatedly taking small steps in the direction that minimises loss.



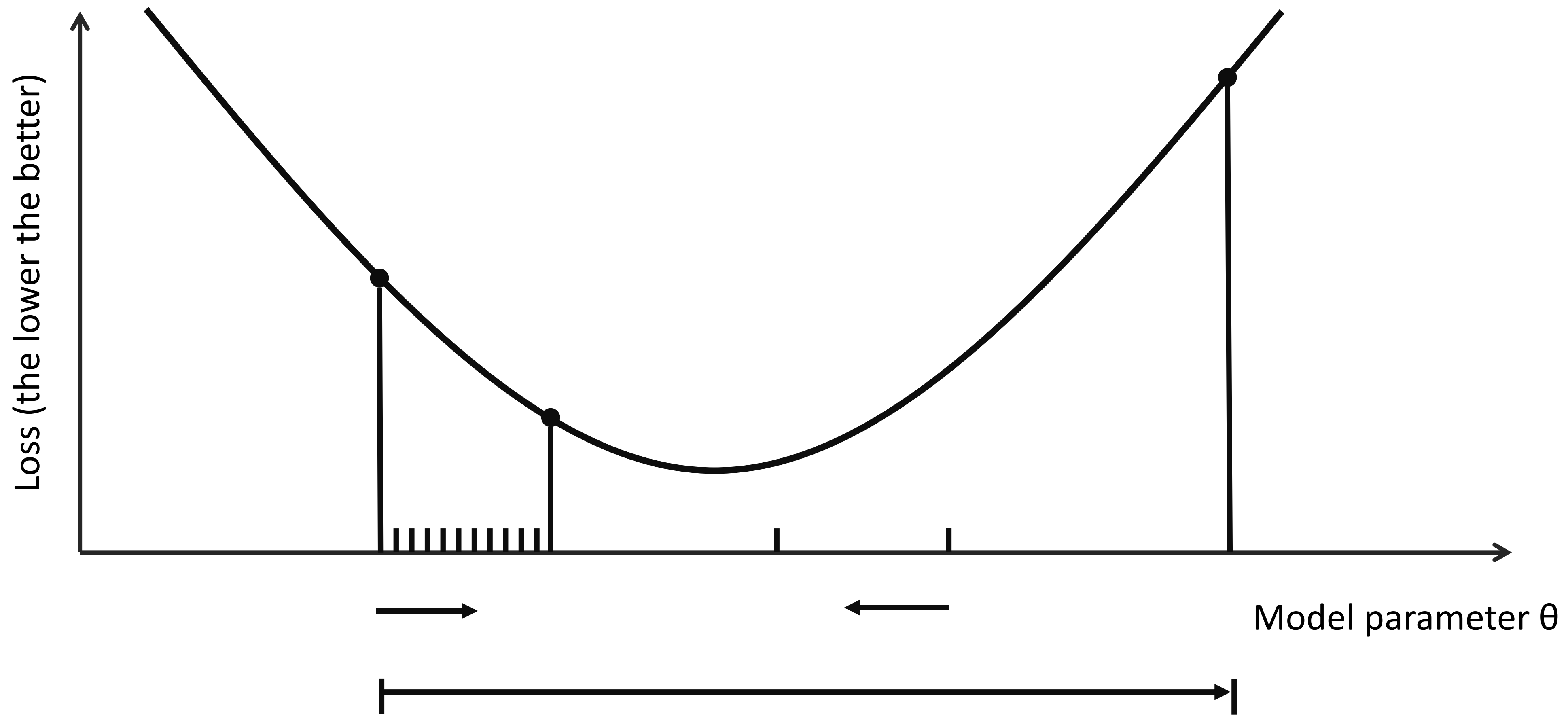
Last Lecture

Gradient Descent

- **Stochastic Gradient Descent**: one example at a time
- **Mini-Batch Gradient Descent**: batches of 10 – 1000
    - Loss & gradients are averaged over the batch

Last Lecture

Learning Rate



Today

- Generalisation
- Training & Test Set
- Representation

Generalisation

Generalisation

How can we make sure that our models are **not over-fit** in practice?

Generalisation

The big picture

- **Goal**: to predict well on new data drawn from (hidden) true distribution.
- **Issue**: we don't see the truth, but we only get to sample from it.
- If it fits current sample well, how can we trust it will predict well on other new samples?

Generalisation

How Do We Know If Our Model Is Good?

- **Theoretically**
  - Interesting field: generalisation theory
  - Based on ideas of measuring model simplicity / complexity
- **Intuition: formalisation of Ockham's Razor principle**
  - The less complex a model is, the more likely a good empirical result is; not just due to the peculiarities of the sample.

Generalisation

How Do We Know If Our Model Is Good?

- **Empirically:**
  - Asking: will our model do well on a new sample of data?
  - Evaluate: get a new sample of data-call it the test set.
  - Good performance on the test set is a useful indicator of good performance on the new data in general:
    - If the test set is large enough.
    - If we don't cheat by using the test set over and over.

Generalisation

The ML Fine Print

Three basic assumptions in all of the above:

1. We draw examples independently and identically (i.i.d.) at random from the distribution.
2. The distribution is stationary - it doesn't change over time.
3. We always pull from the same distribution, including training, validation, and test sets.

Violation of assumptions?

Today

- Generalisation
- Training & Test Set
- Representation

Training & Test Set

Training & Test Set

Partitioning Data Sets

Now how large do we make our different splits?

The larger Training Set

the better model we will be able to learn

The larger Test Set

the better we will be able to have confidence in evaluation metrics, and tighter confidence intervals.

Training & Test Set

What If We Only Have One Data Set?

- **Divide into two sets:**
  - Training set
  - Test set
- **Do not train on test data**
  - Getting surprisingly low loss?
  - Before celebrating, check if you're accidentally training on test data

Training & Test Set

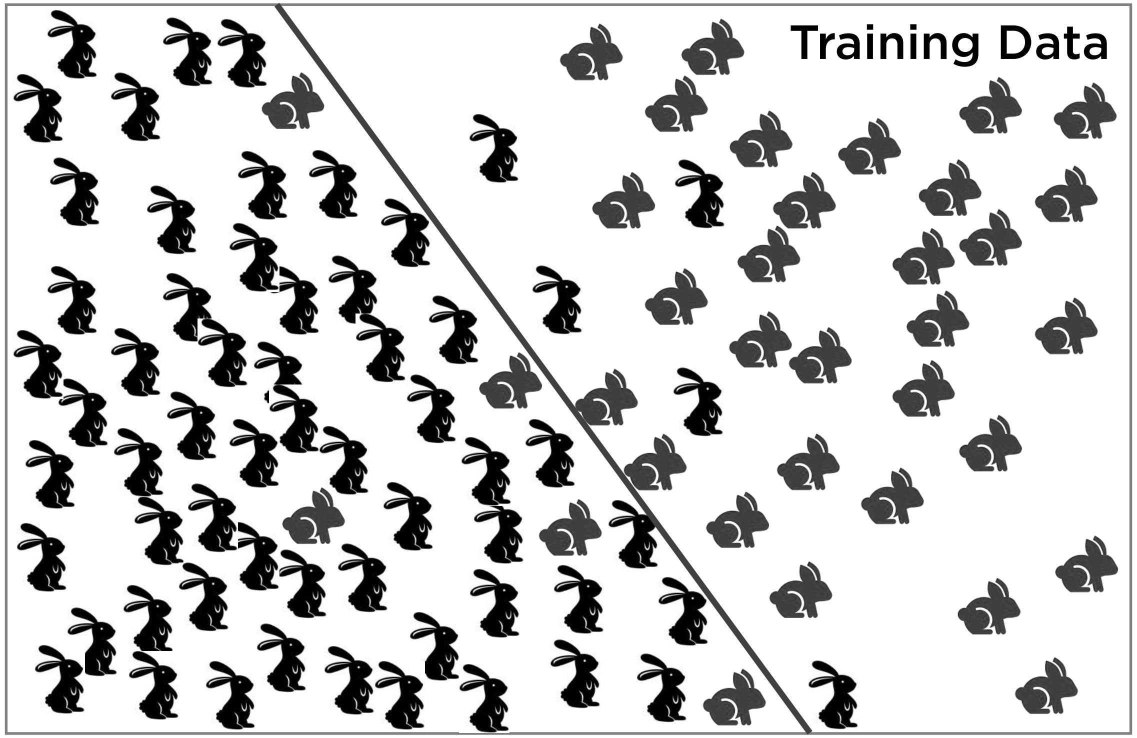
What If We Only Have One Data Set?

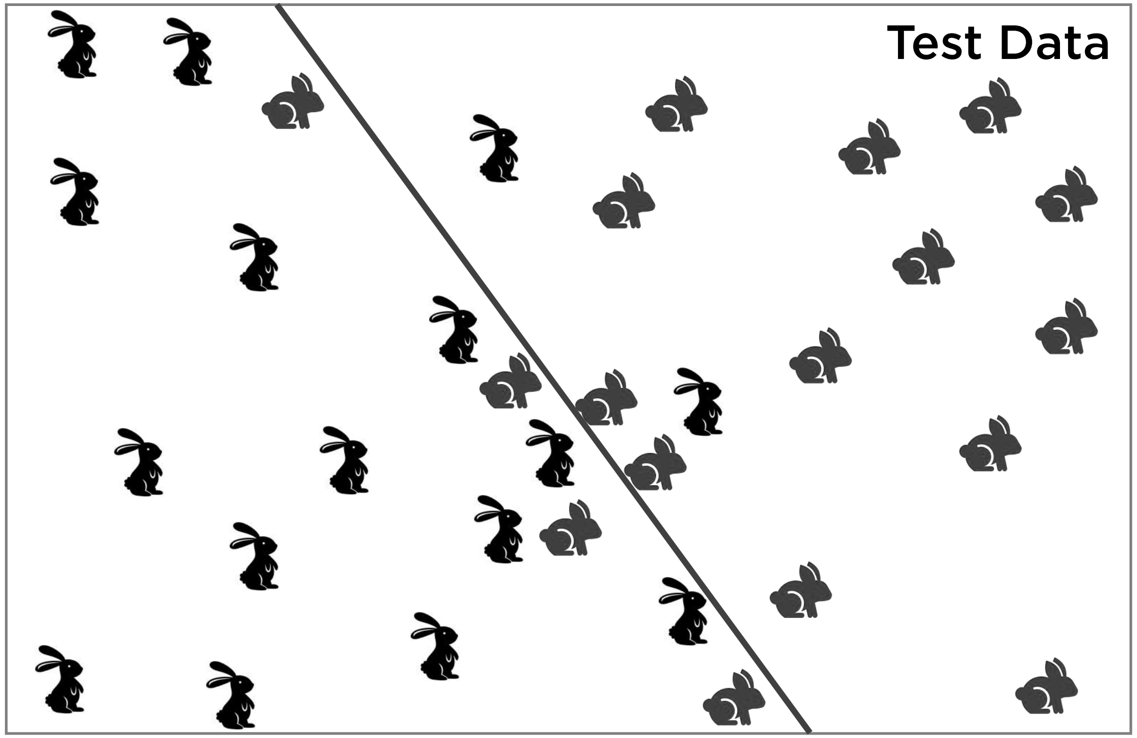
- **Ensure the test set meets the following 2 conditions:**
  - is large enough to yield statistically meaningful results.
  - is representative of the data set as a whole. In other words, don't pick a test set with different characteristics than the training set.

Training & Test Set

What If We Only Have One Data Set?

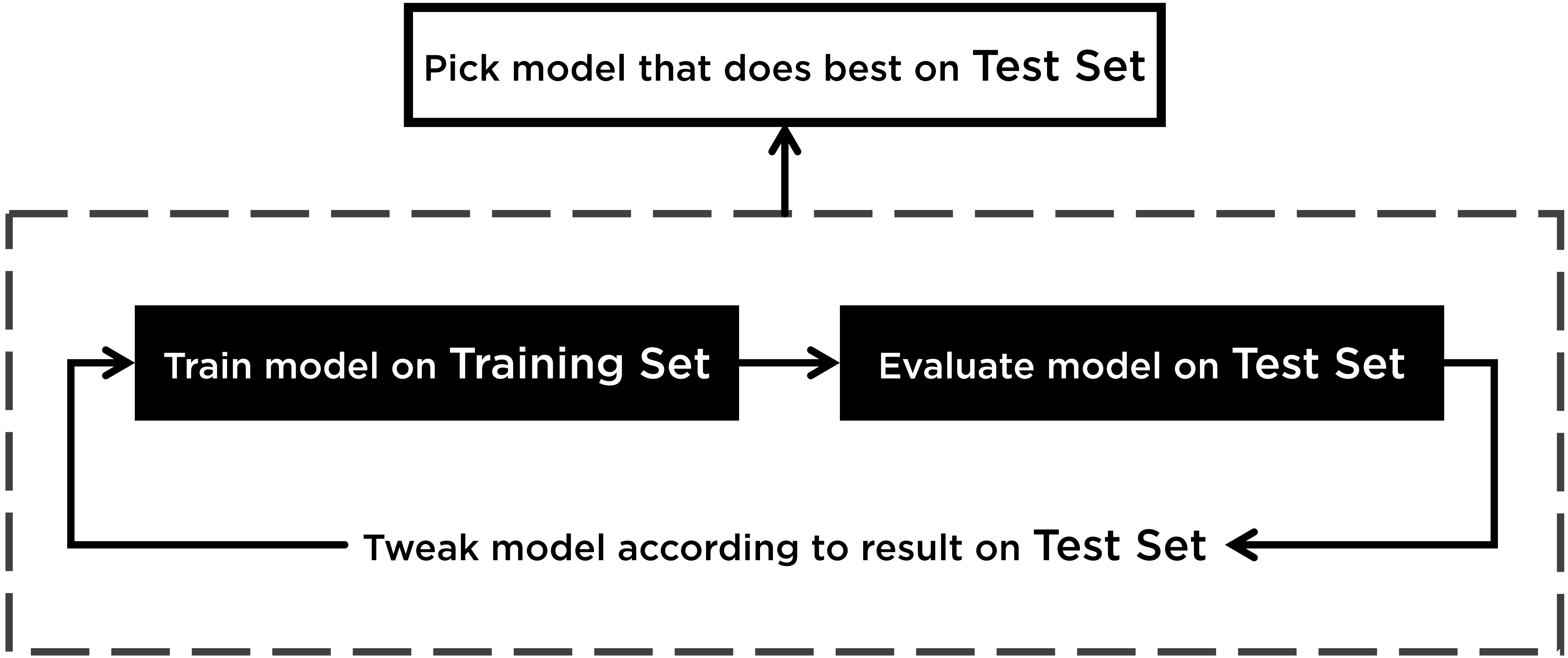
Assuming that your test set meets the preceding two conditions, your goal is to create a model that generalises well to new data. Our test set serves as a proxy for new data.





Training & Test Set

A Possible Workflow?



Training & Test Set

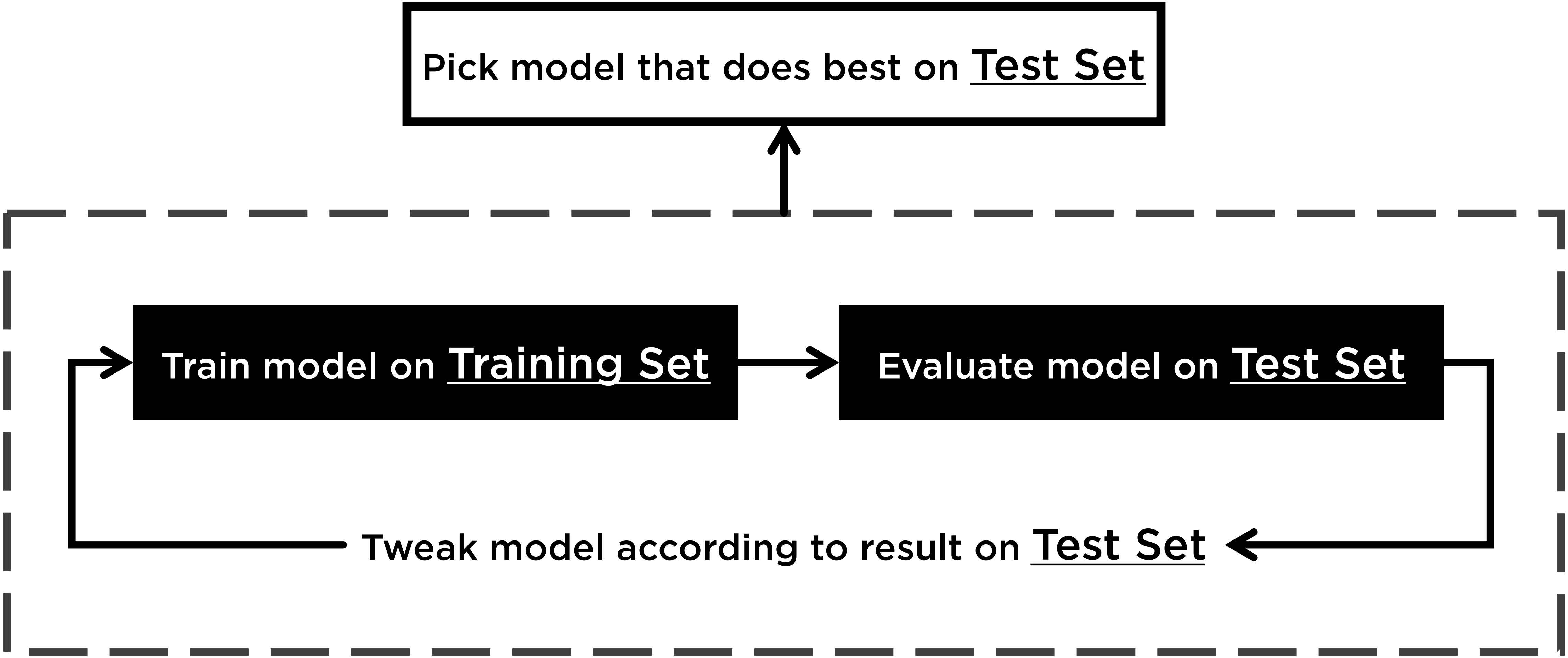
A Possible Workflow?

Training & Test Set

A Possible Workflow?

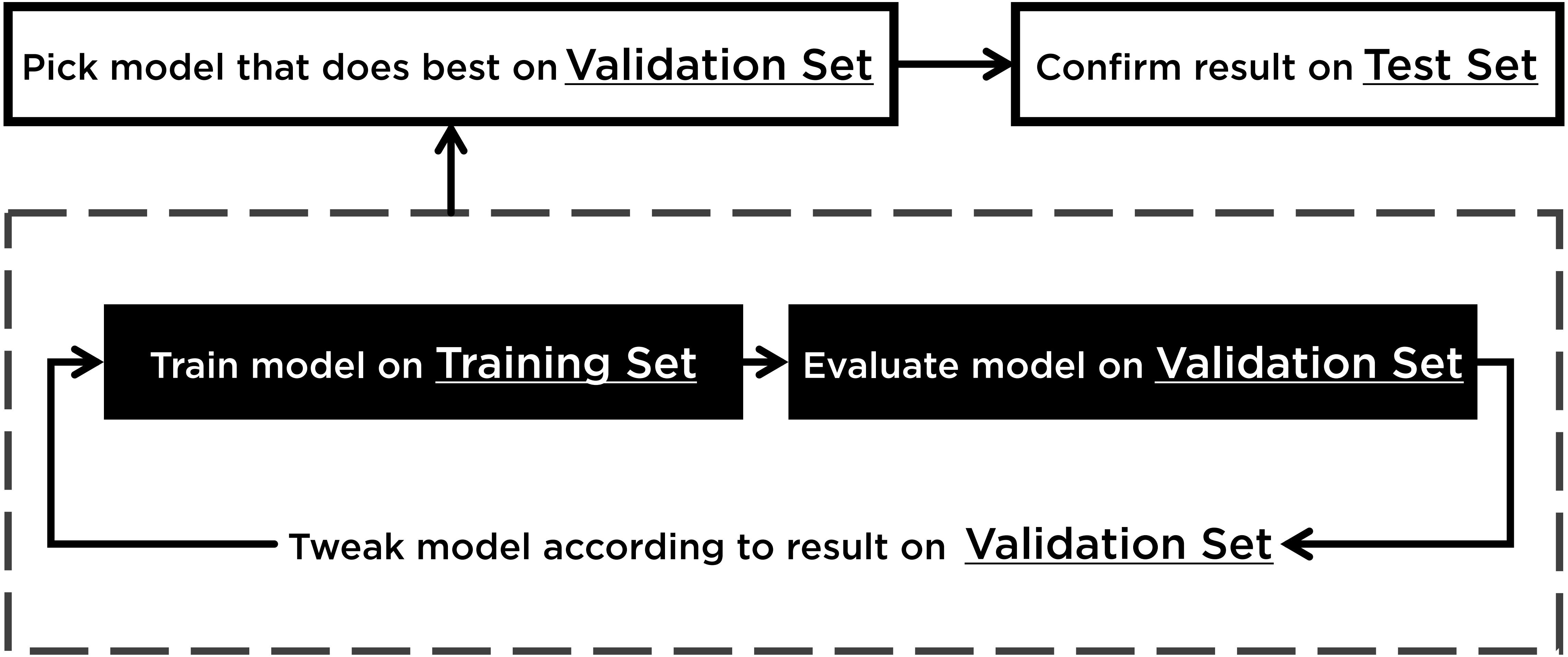
Training & Test Set

A Possible Workflow?



Training & Test Set

Better Workflow: Use a Validation Set



Training & Test Set

Better Workflow: Use a Validation Set

- In this **improved workflow:**
  1. Keeping the **test data** way off to the side (completely unused).
  2. Pick the model that does best on the **validation set**.
  3. Double-check that model against the **test set**.

This is a better workflow because it creates fewer exposures to the **test set**.

Today

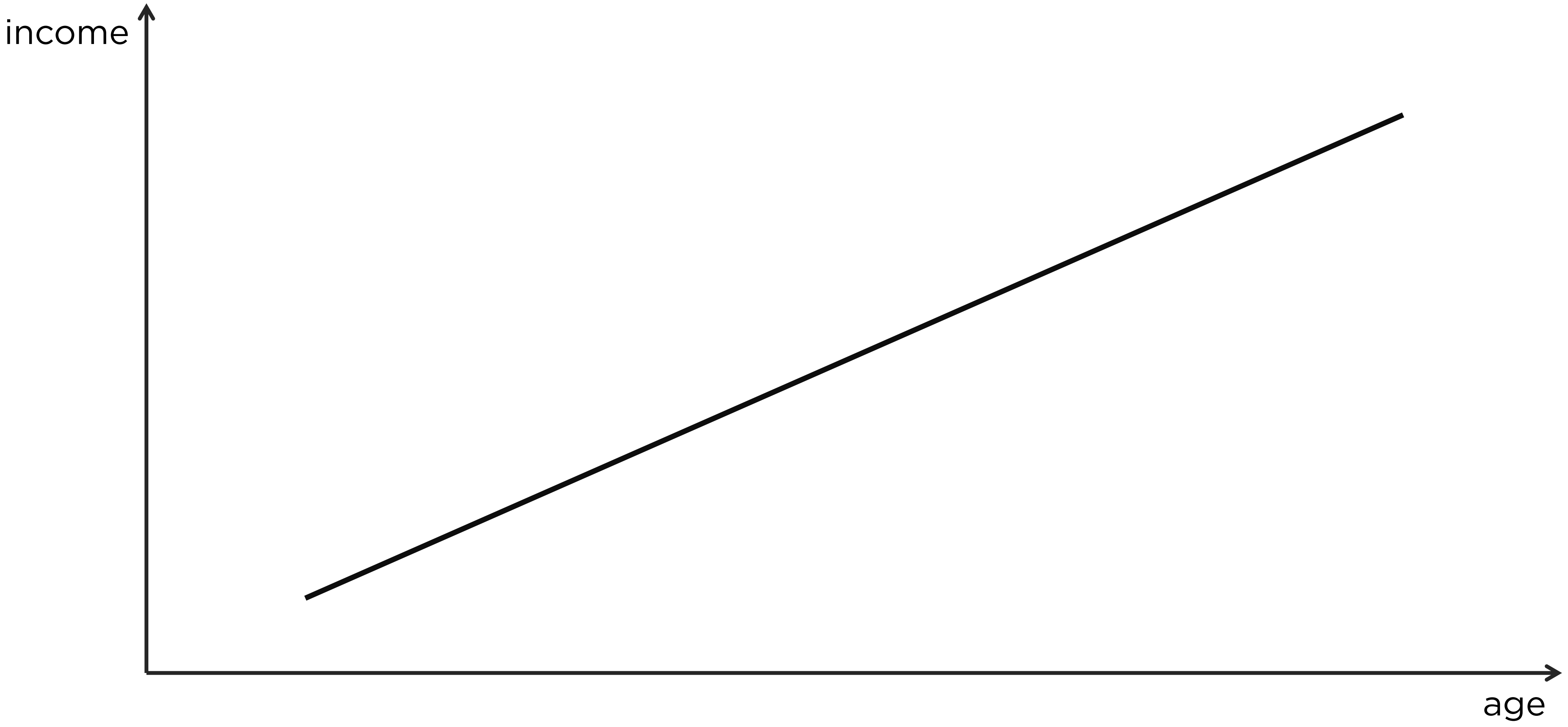
- Generalisation
- Training & Test Set
- Representation

Representation

A machine learning model can't directly see, hear, or sense input examples. Instead, we must create a **Representation** of the data to provide the model with a useful vantage point into the data's key qualities. That is, in order to train a model, we must choose the set of features that best represent the data.

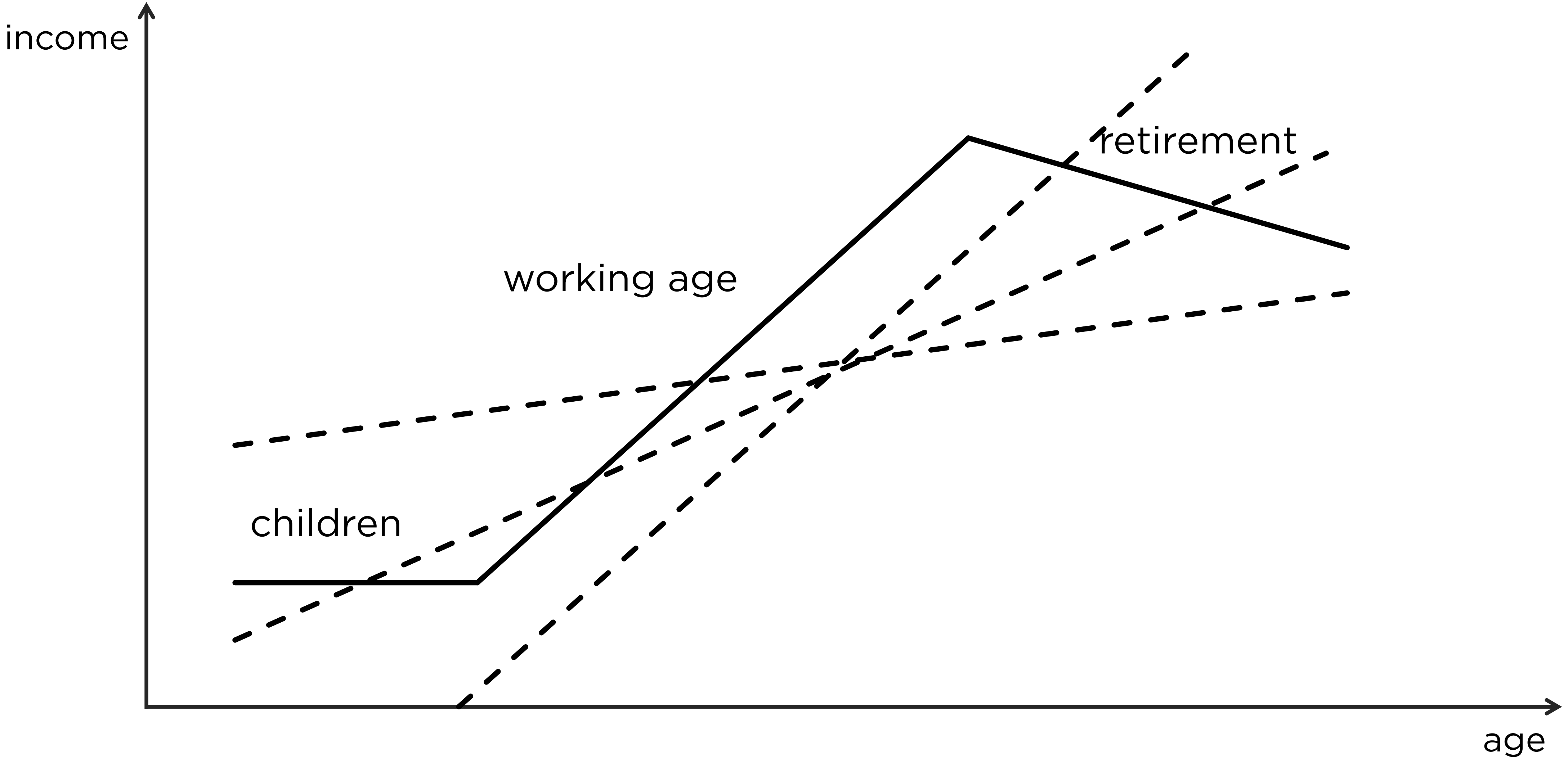
Numeric

Representation - Numeric



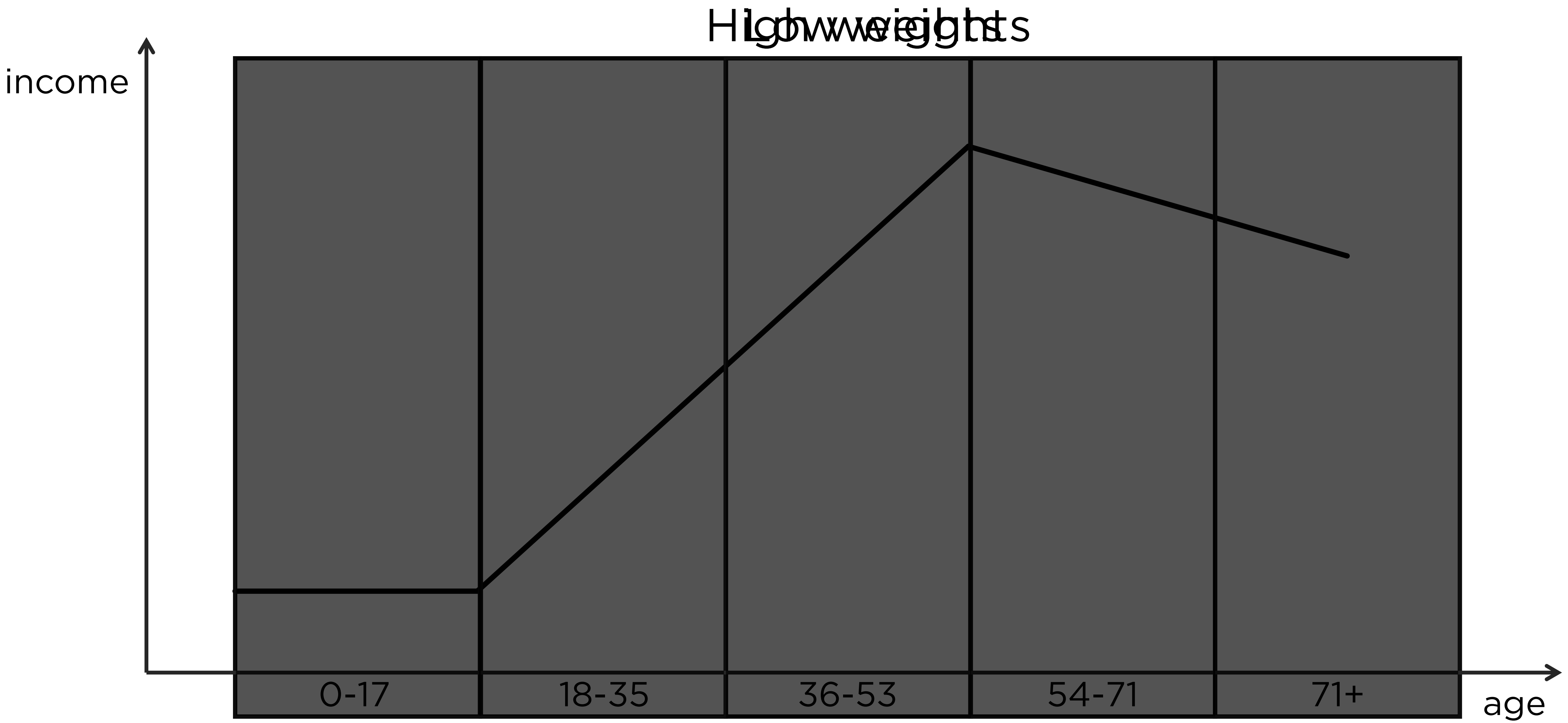
What can go wrong with this approach?

Representation - Numeric



Representation - Numeric

Bucketing - one categorical feature is created for each bucket



Representation - Categorical

Type of blood

A

B

AB

O

Small vocabulary

- use the raw value

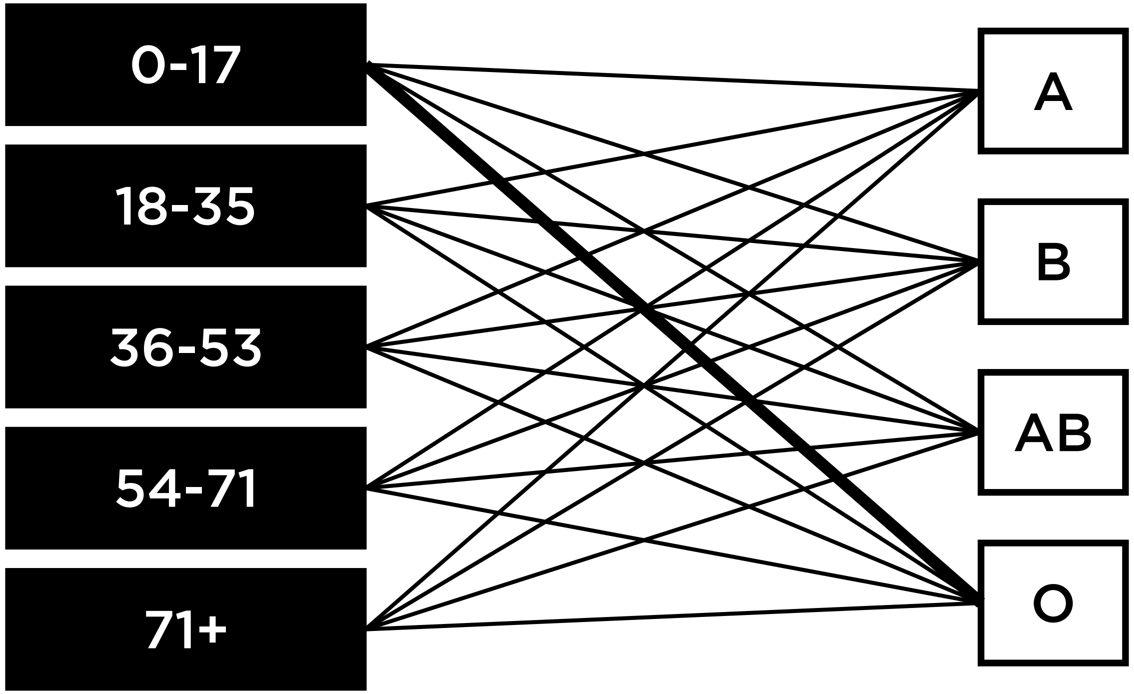
Large vocabulary

- Consider hashing / embedding

Representation - Categorical

Feature Crossing

For each cross, we create a new true/false feature



0-17 & A

0-17 & B

0-17 & AB

0-17 & O

18-35 & A

18-35 & B

...

[0,0,0,1,0,0,...0]

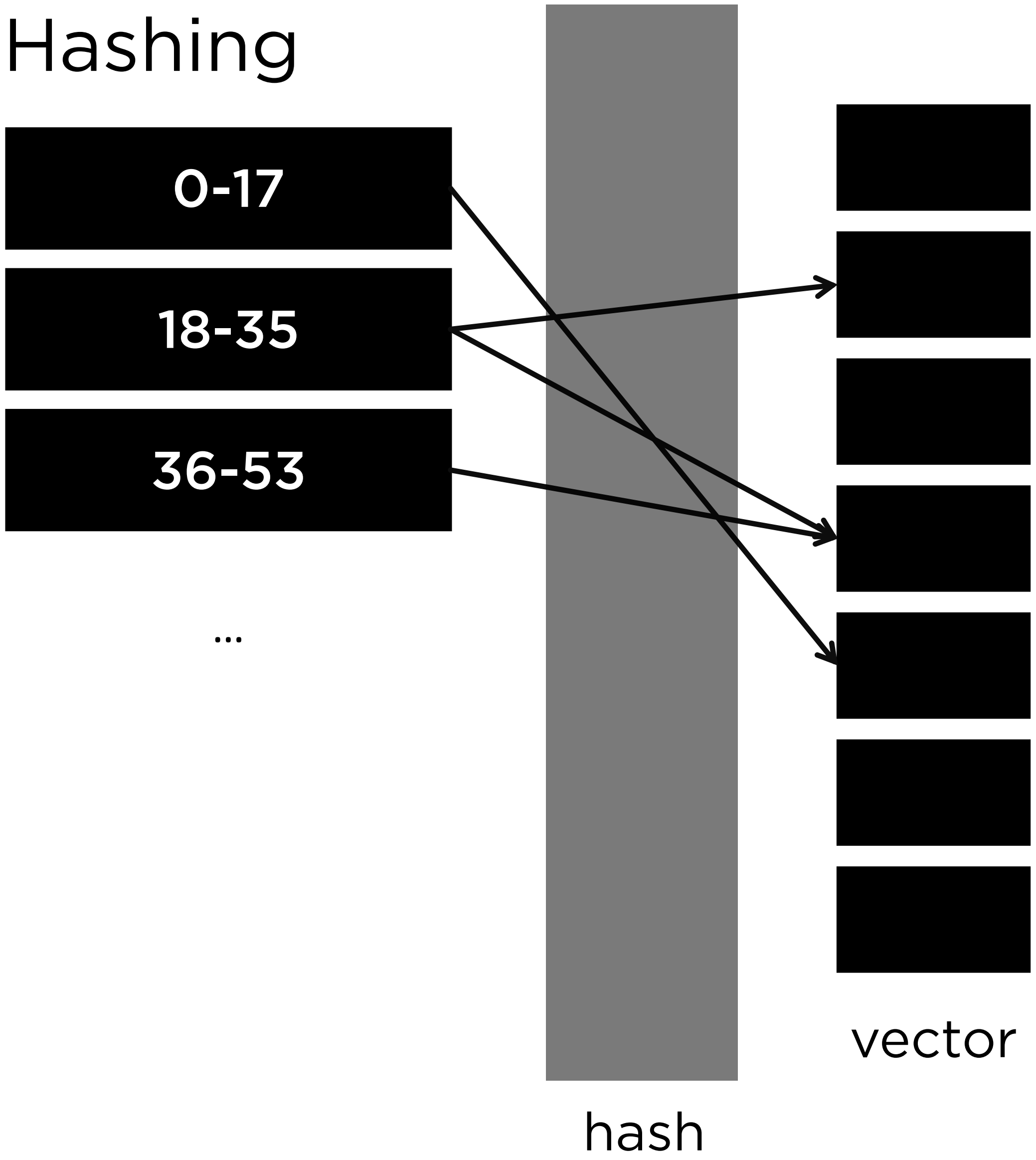
Representation - Categorical

Hashing

- Can save memory
- Can save our time (more importantly!)

Representation - Categorical

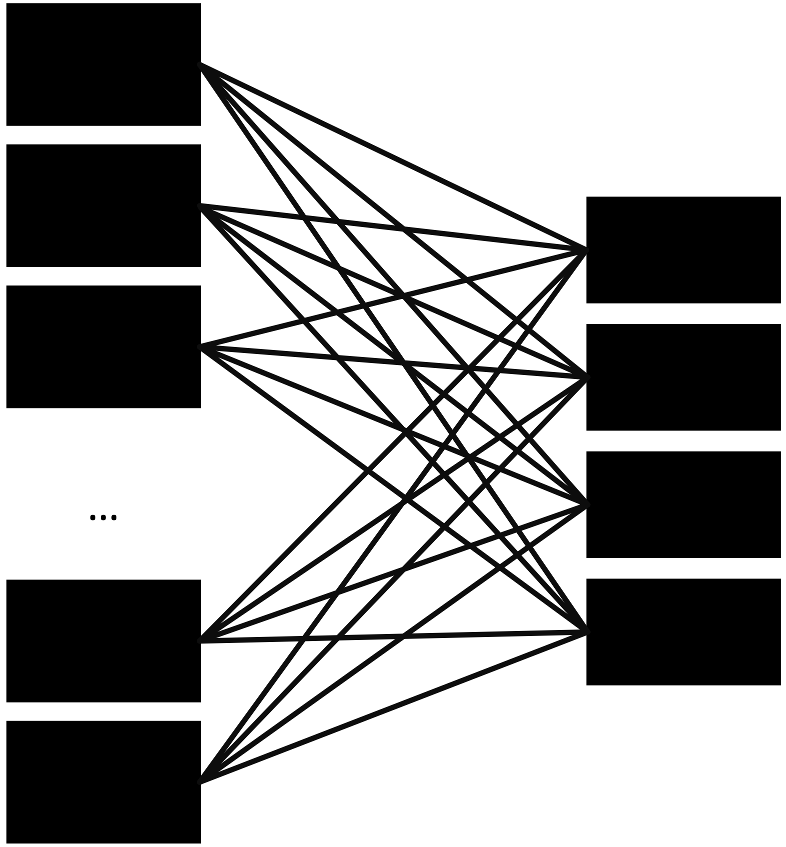
Hashing



- Create a lookup, if we know the vocabulary list in advance.
- Use a hash function to compute automatically, if we don't know the vocab.
- There could be collisions, i.e. different items are mapped to the same value.

Representation - Categorical

Hashing



- Can be used to limit memory usage at the cost of adding some noise to training data.
- Can be used to limit the maximum number of possibilities.

Representation - Categorical

Embedding

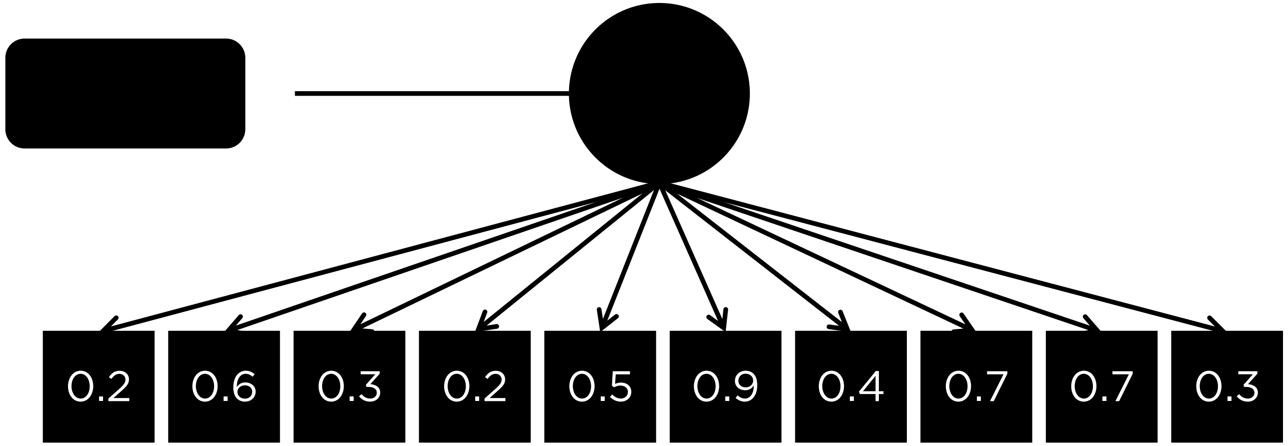
- Powerful way to represent large vocabularies
- Learned automatically
- Dense vectors vs one-hot (sparse)

Representation - Categorical

Embedding - when to use?

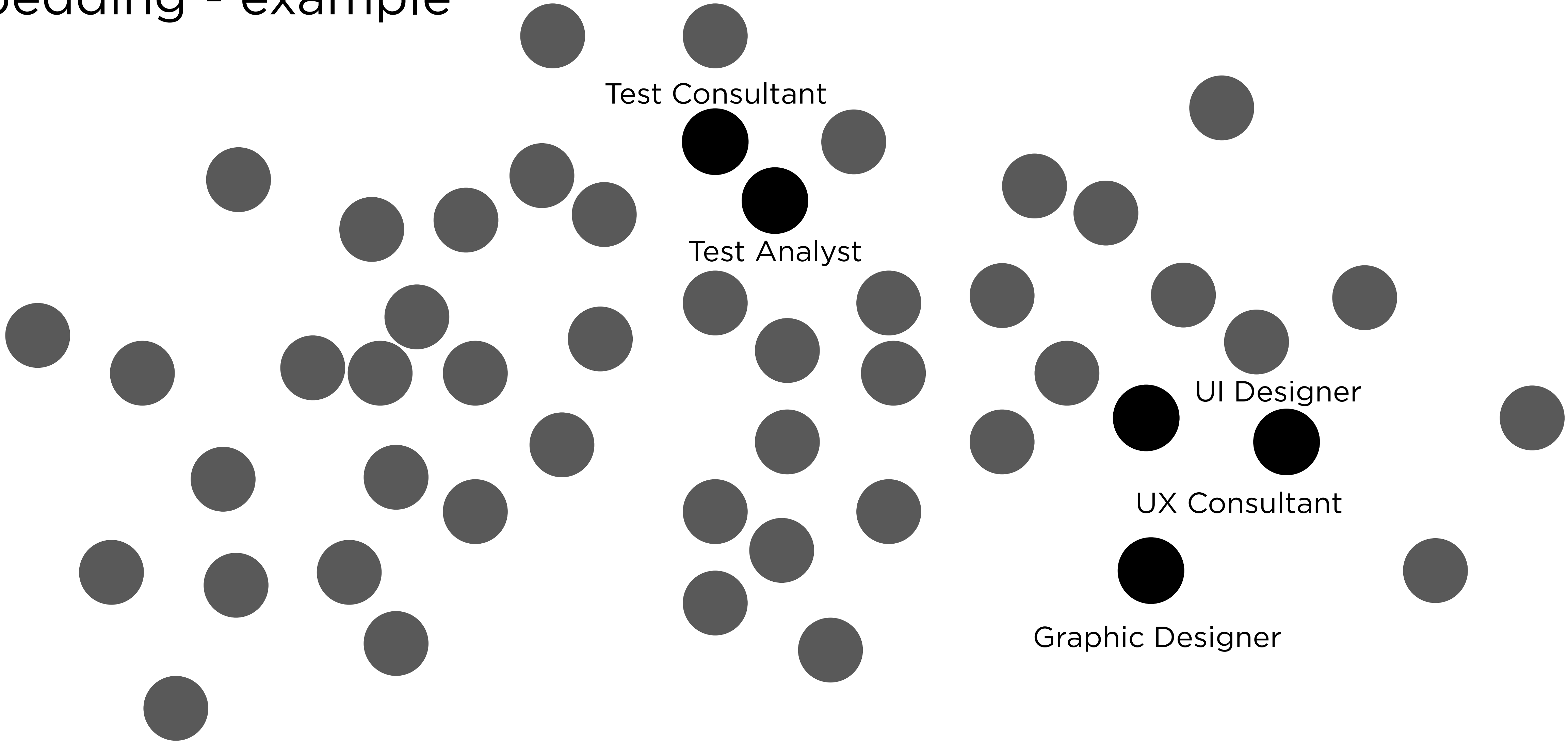
- Large vocabulary
- Concepts (vs specifics)

Embeddings are dense



Representation - Categorical

Embedding - example



Summary

Today

- Generalisation
  - Over-fitting
- Training & Test Set
  - Training and Test Sets
  - Training, Test, and Validation Sets
- Representation
  - Bucketing, Crossing, Hashing, Embedding

Homework

- On DUO – End to End Machine Learning Project

Next Lecture

- Binary Classifier and Performance Measurement