Choosing the Right Machine Learning Problem

Janani Ravi
CO-FOUNDER, LOONYCORN

www.loonycorn.com

Overview

Canonical problems in ML

Classification, regression, clustering, dimensionality reduction

More specialized problem categories

Supervised vs. Unsupervised learning

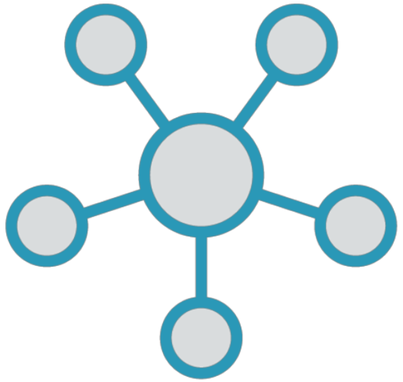
Reinforcement vs. Supervised learning

Choosing the Right Machine Learning Problem

Broad Problem Categories

Classification

Regression



Clustering

Dimensionality reduction

Broad Problem Categories

Classify input data into categories

Regression

Clustering

Dimensionality reduction

Classification Use Cases

Predict categories

Email: spam or ham?

Stocks: Buy, sell or hold?

Images: Cat, dog or mouse?

Text: Positive, negative or neutral sentiment?

Broad Problem Categories

Classification

Regression

Clustering

Dimensionality reduction

Broad Problem Categories

Classification

Predict continuous numeric values

Clustering

Dimensionality reduction

Regression Use Cases

Given past stock data predict price tomorrow

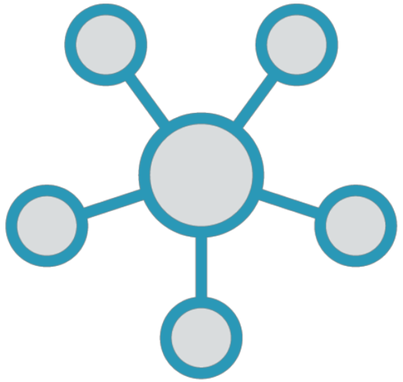
Given characteristics of a car predict mileage

Given location and attributes of a home predict price

Broad Problem Categories

Classification

Regression



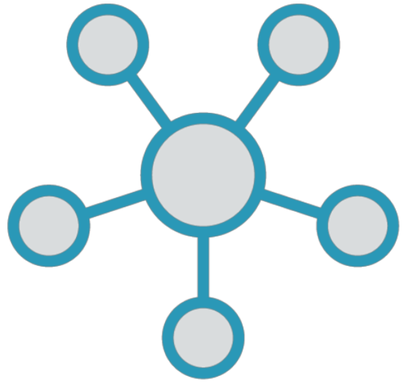
Clustering

Dimensionality reduction

Broad Problem Categories

Classification

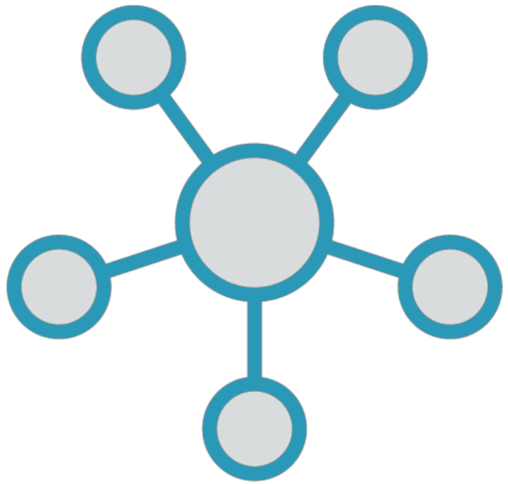
Regression



Discover patterns and groupings in data

Dimensionality reduction

Clustering Use Cases



Document discovery - find all documents related to homicide cases

Social media ad targeting - find all users who are interested in sports

Broad Problem Categories

Classification

Regression

Clustering

Dimensionality reduction

Broad Problem Categories

Classification

Regression

Clustering

Feature Detection

Dimensionality Reduction Use Cases

Find latent drivers of stock movements

Pre-process data to build more robust machine learning models

Improve performance of models in training

Supervised Learning

Classification

Regression

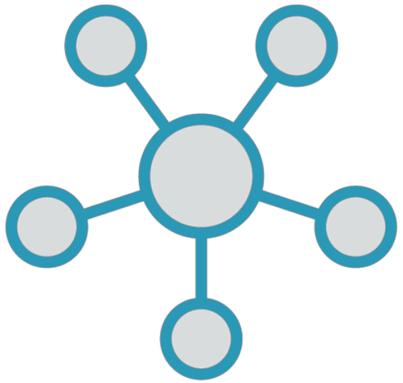
Clustering

Dimensionality reduction

Unsupervised Learning

Classification

Regression



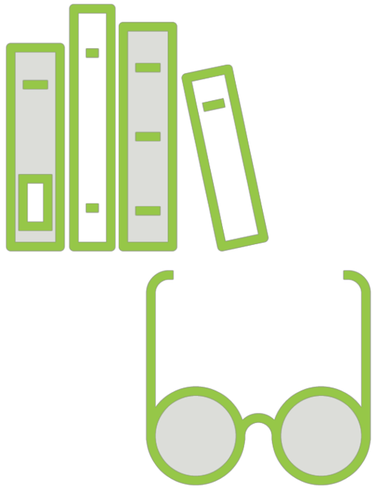
Clustering

Dimensionality reduction

Specialized Problem Categories

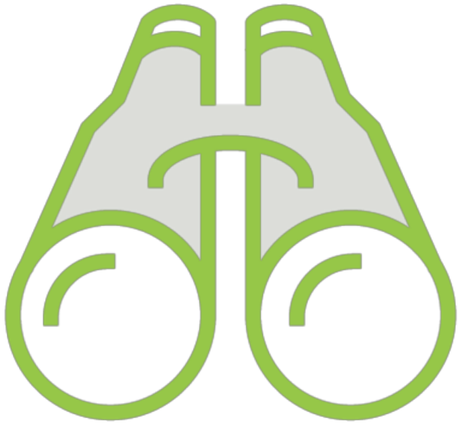
Recommendation Systems

Recommend products to users



Association Rules Detection

Detect transactions that occur together



Reinforcement Learning

Train agent to navigate an uncertain environment

Broad Solution Categories

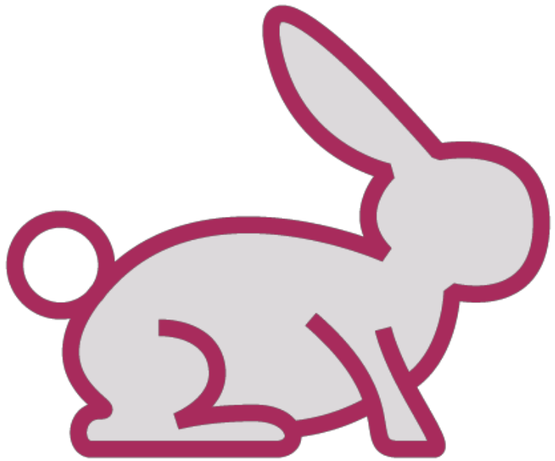
Use-case	Problem
Image data	Convolutional Neural Networks
Complex textual data	Recurrent Neural Networks
Sequential or time series data	Recurrent Neural Networks
Linear x-variables	Linear and logistic regression, PCA
Twisted data (S-curves, Swiss Rolls)	Manifold learning
Large numbers of x-variables	Decision trees

Supervised and Unsupervised Learning

“What lies behind us and what lies ahead of us are tiny matters compared to what lives within us”

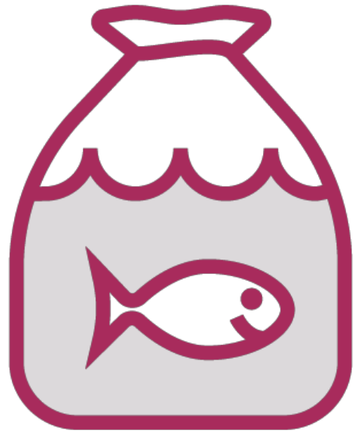
Henry David Thoreau

Whales: Fish or Mammals?



Mammals

Members of the infraorder *Cetacea*



Fish

Look like fish, swim like fish, move with fish

Whales: Fish or Mammals?

ML-based Classifier

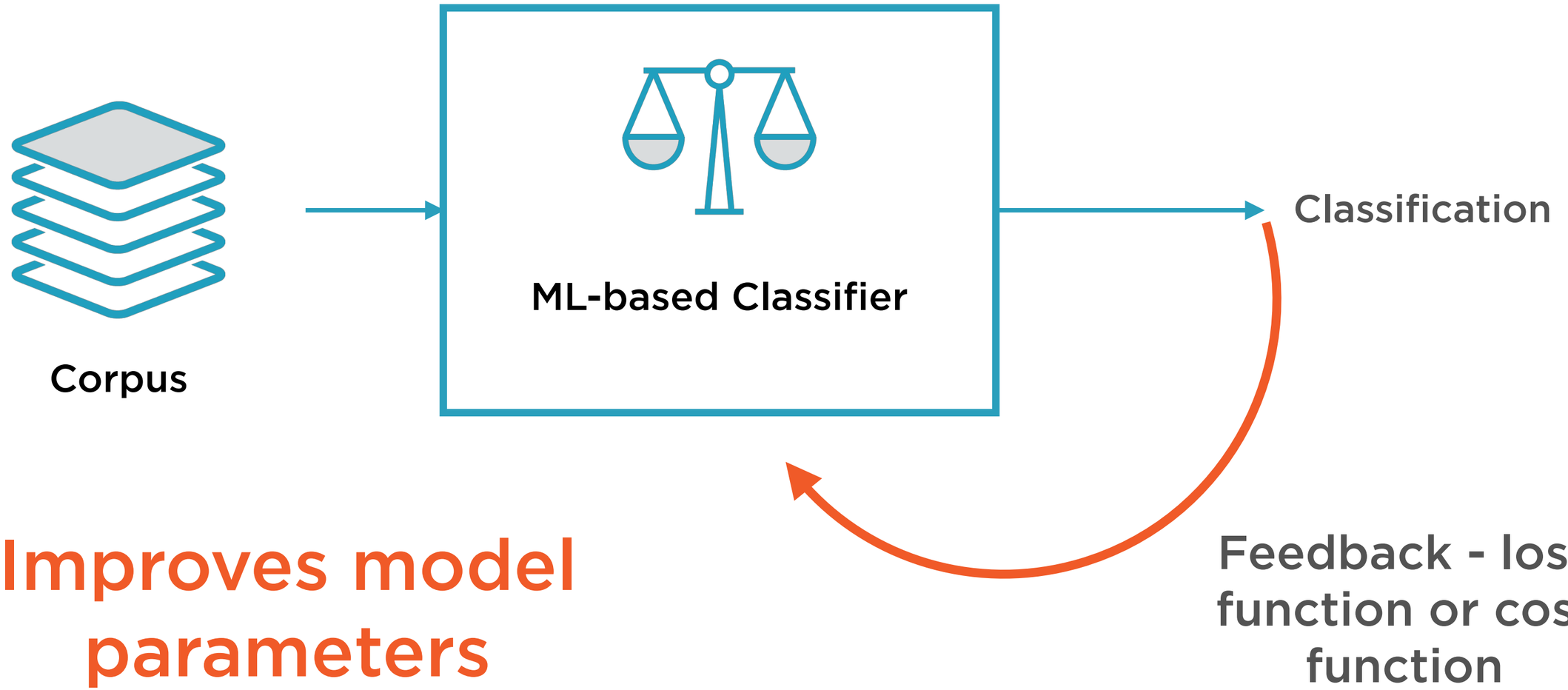
Training

Feed in a large corpus of data classified correctly

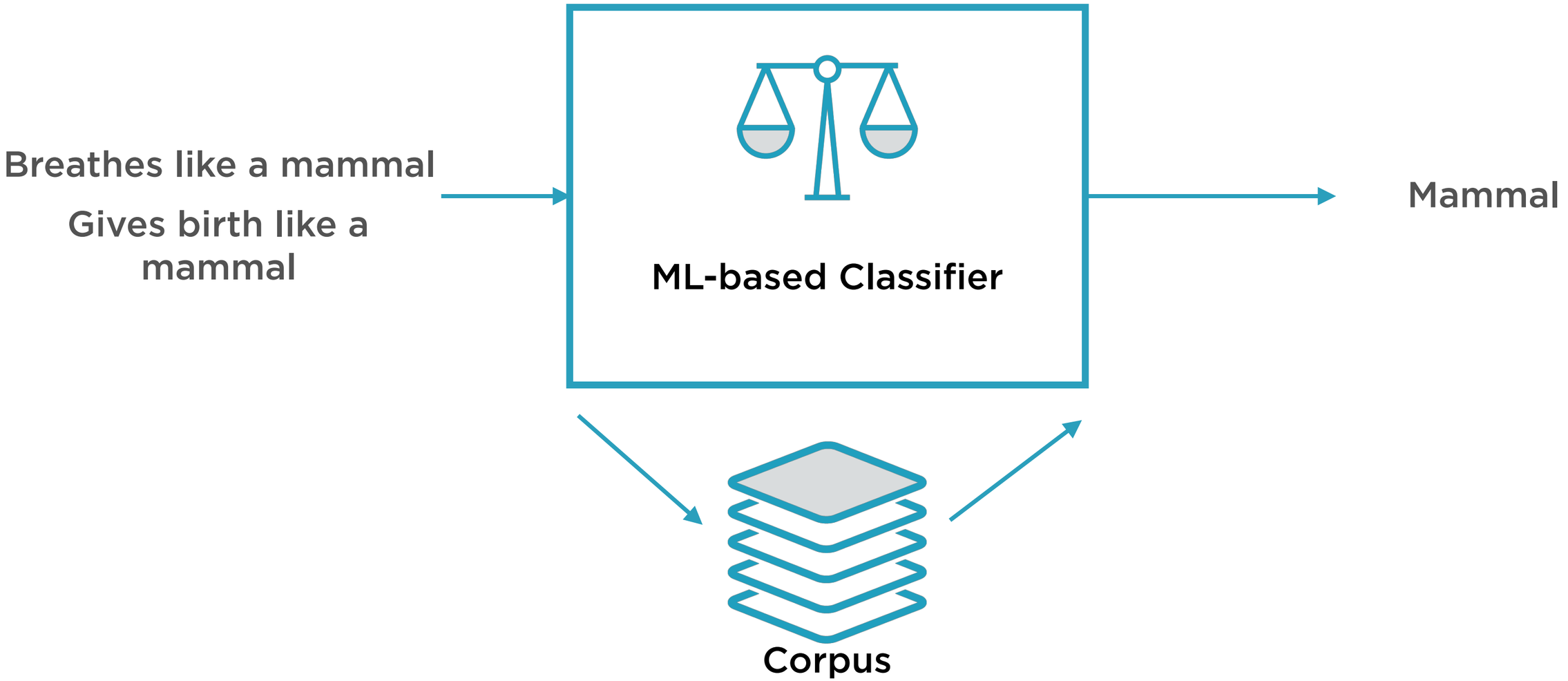
Prediction

Use it to classify new instances which it has not seen before

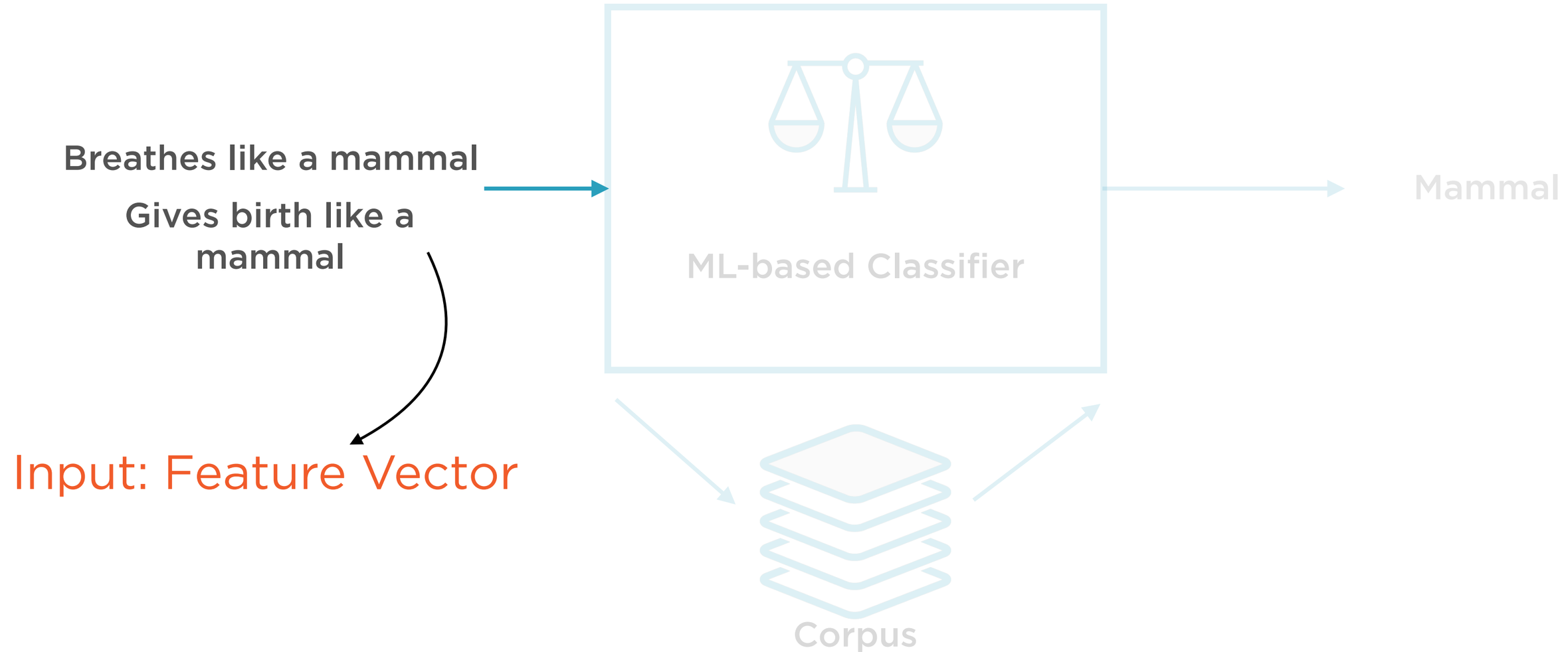
Training the ML-based Classifier



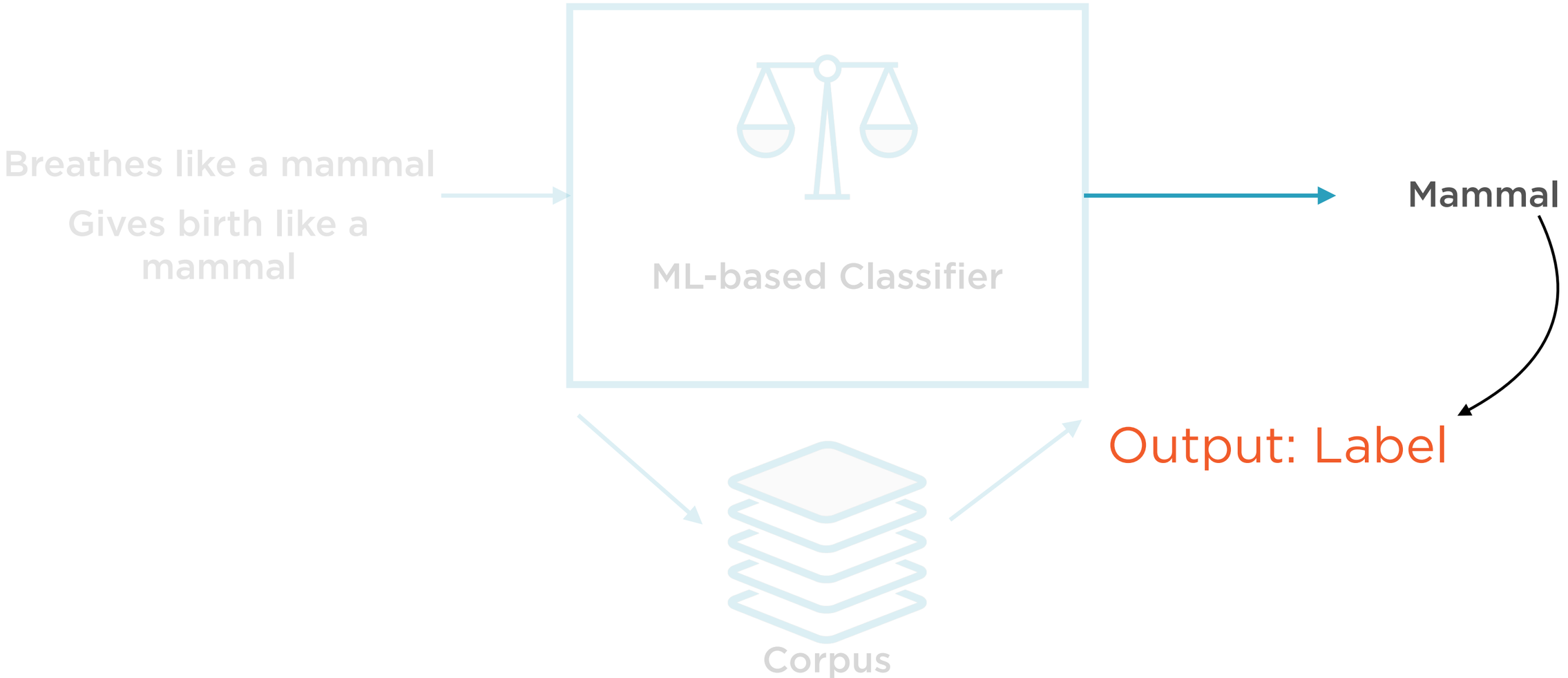
“Traditional” ML-based Binary Classifier



“Traditional” ML-based Binary Classifier



“Traditional” ML-based Binary Classifier



```
y = f(x)
```

Supervised Machine Learning

Most machine learning algorithms seek to “learn” the function f that links the features and the labels

```
y = Wx + b
```

$$f(x) = Wx + b$$

Linear regression specifies, up-front, that the function f is linear

```
def doSomethingReallyComplicated(x1,x2…):
   …
   …
   …
   return complicatedResult
```

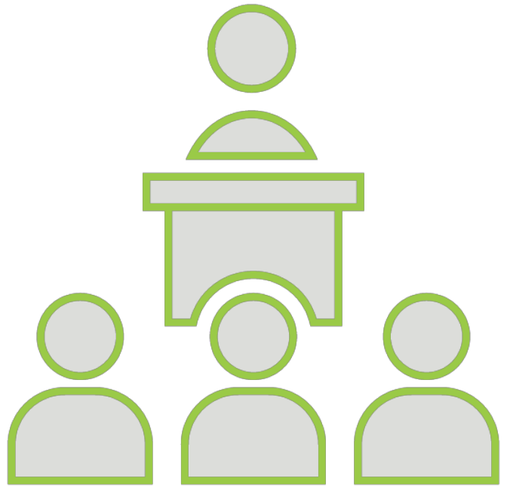
---

f(x) = doSomethingReallyComplicated(x)

ML algorithms such as neural network can “learn” (reverse-engineer) pretty much anything given the right training data

Unsupervised Learning learns patterns in data **without a labeled corpus**

Types of ML Algorithms



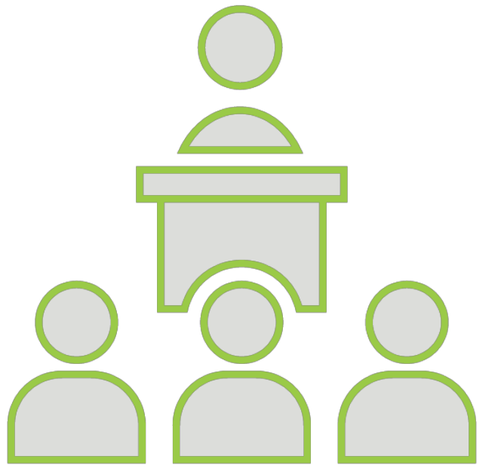
Supervised

Labels associated with the training data is used to correct the algorithm

Unsupervised

The model has to be set up right to learn structure in the data

Supervised Learning



Input variable x and output variable y

Learn the mapping function y = f(x)

Approximate the mapping function so for new values of x we can predict y

Use existing dataset to correct our mapping function approximation

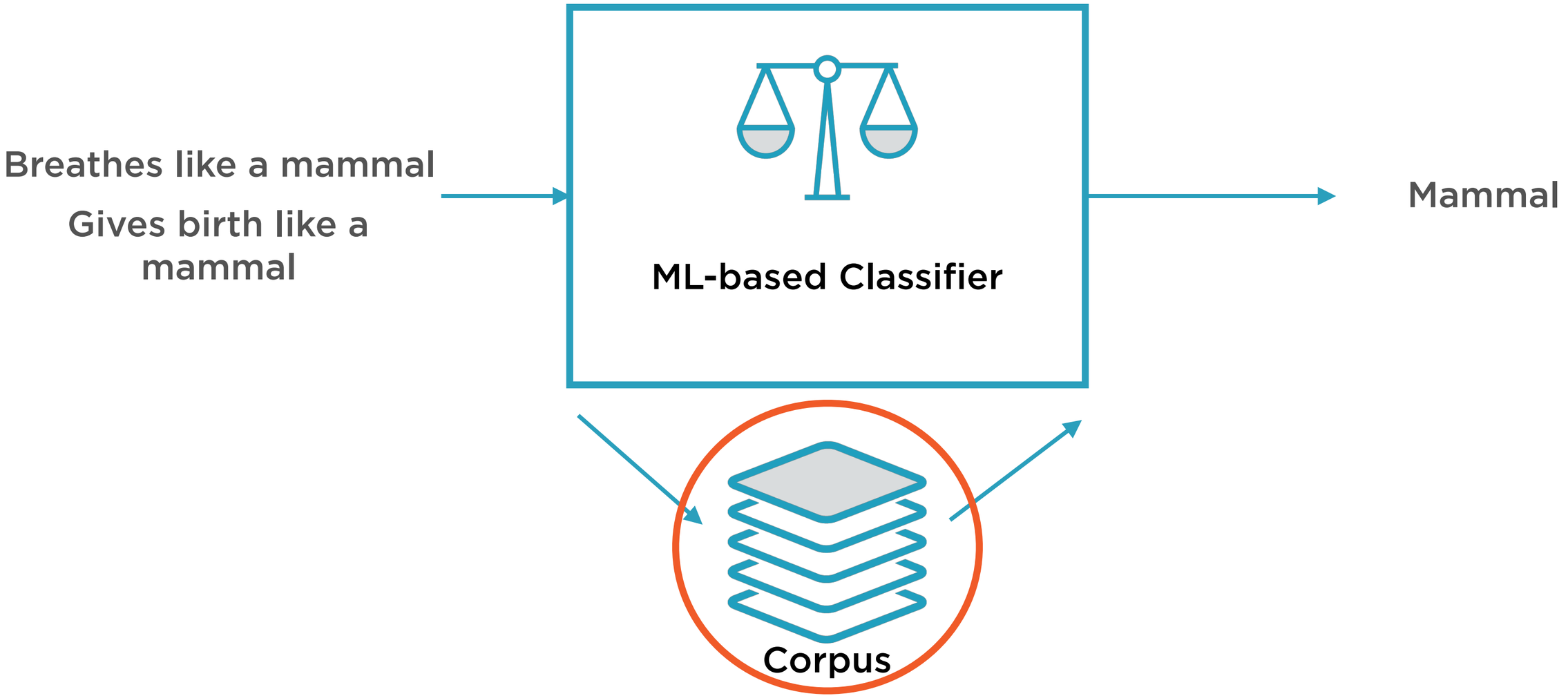
Unsupervised Learning

Only have input data x – no output data

Model the underlying structure to learn more about data

Algorithms self discover the patterns and structure in the data

No Labelled Training Data



Why Look Within?

To be emotionally self-sufficient

To learn what values matter (to you)

Identify others who share them…

..and those who don’t

Eliminate what does not matter

In general, to train yourself to navigate the outside world

Why Look Within?

In Life	In Machine Learning
To be emotionally self-sufficient	To make unlabelled data self-sufficient
To learn what values matter to you	Latent factor analysis
Identify others who share them…	Clustering
..and those who don’t	Anomaly detection
Eliminate what does not matter	Quantization
In general, to train yourself to navigate the outside world	Pre-training for supervised learning problems (classification, regression)

Unsupervised Learning Use-cases

ML Technique	Use-case
To make unlabelled data self-sufficient	Identify photos of a specific individual
Latent factor analysis	Find common drivers of 200 stocks
Clustering	Find relevant document in a corpus
Anomaly detection	Flag fraudulent credit card transactions
Quantization	Compress true color (24 bit) to 8 bit
Pre-training for supervised learning problems (classification, regression)	All of the above!

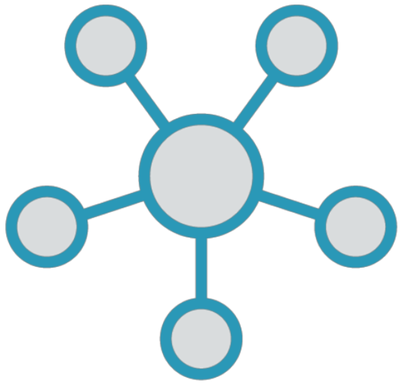
Unsupervised Learning Use-cases

What	How
To make unlabelled data self-sufficient	Autoencoder
Latent factor analysis	Autoencoder
Clustering	Clustering
Anomaly detection	Autoencoder
Quantization	Clustering
Pre-training for supervised learning problems (classification, regression)	All of the above!

Unsupervised Learning

Classification

Regression



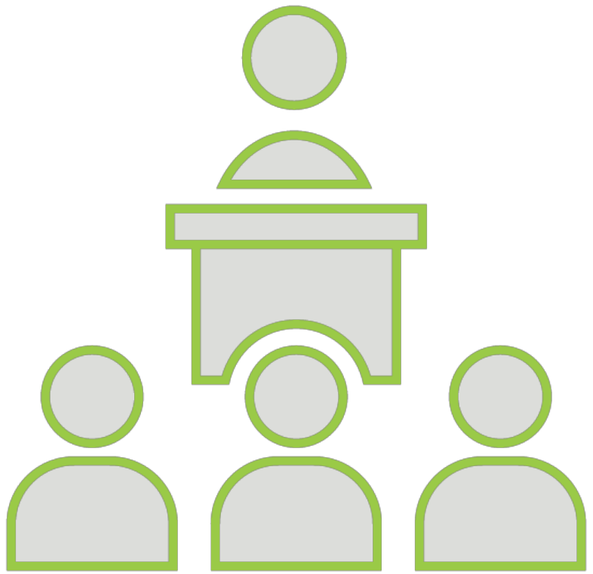
Clustering

Dimensionality reduction

Reinforcement Learning

“If you ask the wrong question, you will never get the right answer”

Supervised and Unsupervised Learning



Supervised

Given x, predict y

Unsupervised

Given x, simplify x

Neither supervised nor unsupervised learning will work in an **unknown** environment

Reinforcement Learning

Train decision makers to take actions to maximize rewards in an uncertain environment

Reinforcement Learning

Train decision makers to take actions to maximize rewards in an uncertain environment

Software decision makers i.e. programs or agents

Reinforcement Learning

Train decision makers to take actions to maximize rewards in an uncertain environment

The output of the program is a set of actions, rather than a set of predictions

Reinforcement Learning

Train decision makers to take actions to maximize rewards in an uncertain environment

The algorithm that determines these actions is called the **policy**

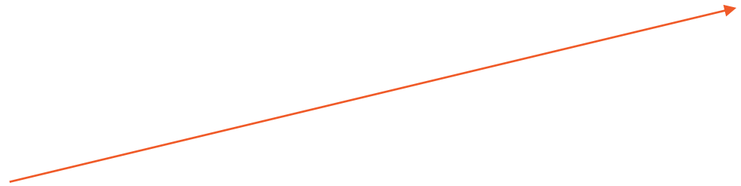
Reinforcement Learning

Train decision makers to take actions to maximize rewards in an uncertain environment

Those actions must be optimized to earn rewards (and avoid punishments)

Reinforcement Learning

Train decision makers to take actions to maximize rewards in an uncertain environment



Those rewards and punishments are externally imposed (by the environment)

Reinforcement Learning

Train decision makers to take actions to maximize rewards in an uncertain environment

The environment is complex, so the reward/ punishment for actions is usually not known in advance

Reinforcement Learning

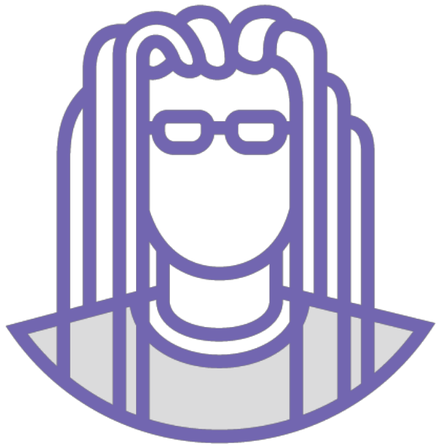
Train decision makers to take actions to maximize rewards in an uncertain environment

The decision maker (program) needs to be trained to explore that uncertain environment - combining caution and courage

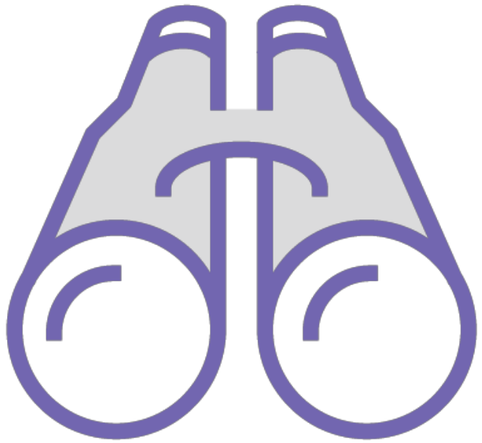
Reinforcement Learning

Train decision makers to take actions to maximize rewards in an uncertain environment

Reinforcement Learning

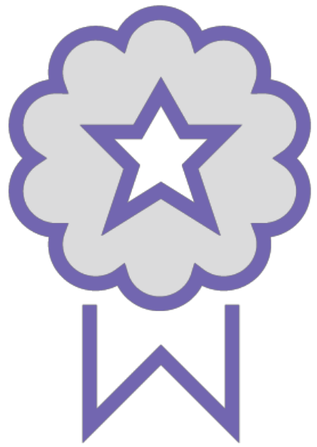


Agent - the decision maker in an environment



Observes the environment

Takes actions



Gets rewards

Reinforcement Learning

Agent - the decision maker in an environment

Observes the environment

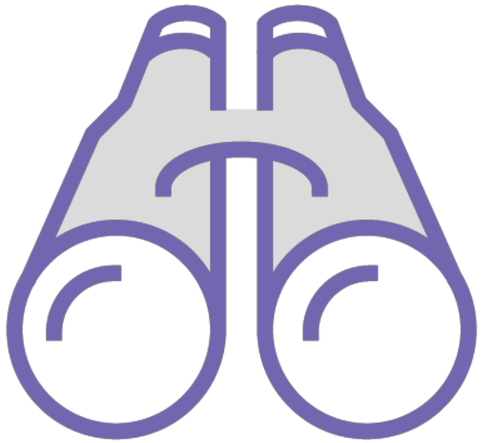
Takes actions

Gets rewards

The policy determines the action

Reinforcement Learning

Agent - the decision maker in an environment



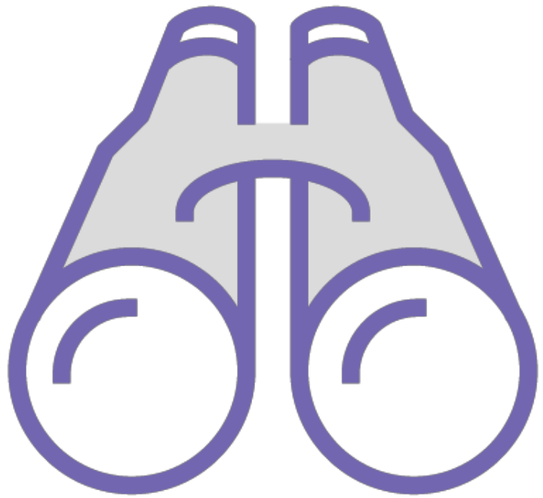
Observes the environment

Takes actions

Gets rewards

Policy is determined by exploring the environment using an algorithm

Reinforcement Learning Use Cases



Robotics: Self-navigating robots

Text mining: Generating summaries of text data

Healthcare: Optimizing medication dosing

Reinforcement vs. Supervised/Unsupervised

Reinforcement Learning

Objective: choose "best" actions

Environment is uncertain

Supervised/Unsupervised Learning

Objective: Predict, classify or simplify

Environment is known (x is known)

Reinforcement vs. Supervised/Unsupervised

Reinforcement Learning

Training involves exploring the environment

Wrong actions get punished, right actions get rewarded

Training process involves determining the "best" policy

Explicit dependency of rewards on previous actions

Supervised/Unsupervised Learning

Training involves finding patterns in data or is entirely absent

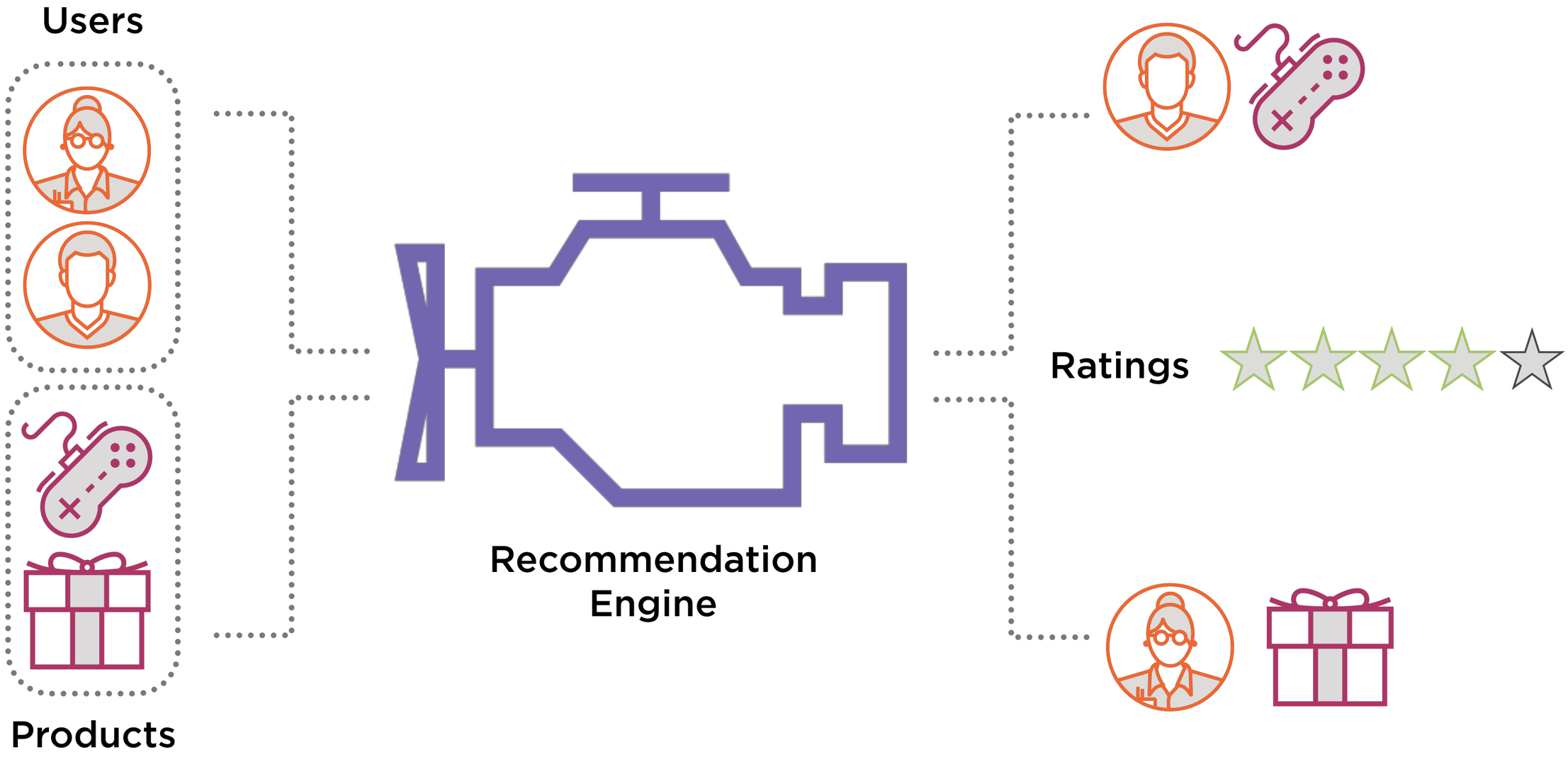
Loss of incorrect predictions used to train model

Training process involves fitting the "best" model

Individual points are independent of each other

Recommendation Systems

Recommendation Systems



Approaches to Recommendations

Content-based

Estimate rating using this user and this product alone

Collaborative

Employ information about other users, products too

Hybrid

Combine both content-based and collaborative filtering

Approaches to Recommendations

Content-based

Estimate rating using this user and this product alone

Collaborative

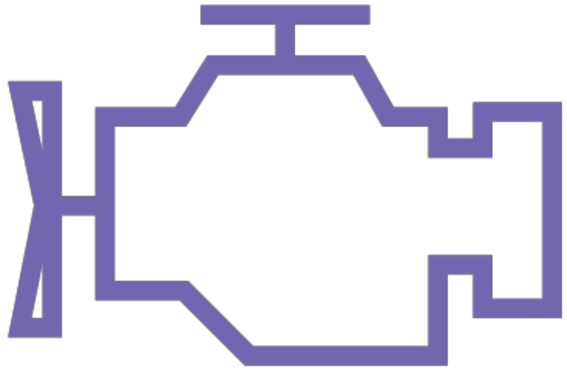
Employ information about other users, products too

Hybrid

Combine both content-based and collaborative filtering

Content-based Filtering

Content-based Filtering



Match product description to user profile

Two significant drawbacks

- Requires accurate, rich product metadata
- Hard to extend across product types

Approaches to Recommendations

Content-based

Estimate rating using this user and this product alone

Collaborative

Employ information about other users, products too

Hybrid

Combine both content-based and collaborative filtering

Collaborative Filtering

Collaborative Filtering

Individual Users

Products

Personalized Recommendations

Views

Aggregate of Users

Personalized Recommendations

Purchases

Collaborative Filtering

Individual Users

Products

Personalized Recommendations

Views

Personalized Recommendations

Purchases

Aggregate of Users

Collaborative Filtering

Users who agreed in the past will agree in the future, and that they will like similar kinds of items as they liked in the past

Collaborative Filtering

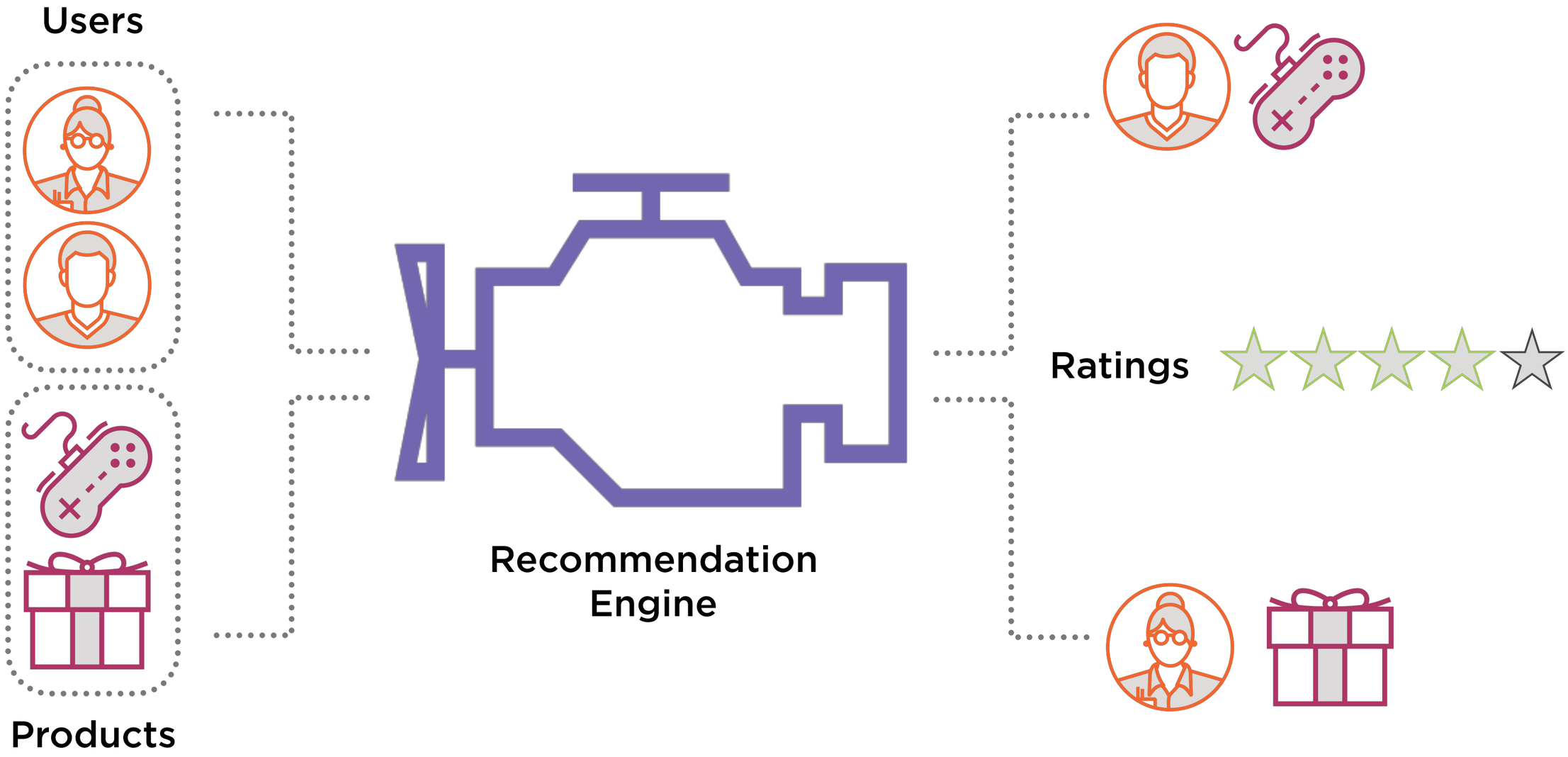
Users who agreed in the past will agree in the future, and that they will like similar kinds of items as they liked in the past

Collaborative Filtering

Users who agreed in the past will agree in the future, and that they will like similar kinds of items as they liked in the past

“People who buy X also buy Y”

Recommendation Systems



Estimate how a user would rate every product

Recommend the products to the user which have the highest estimated ratings

Summary

Canonical problems in ML

Classification, regression, clustering, dimensionality reduction

More specialized problem categories

Supervised vs. Unsupervised learning

Reinforcement vs. Supervised learning