Machine Learning Foundations

Sara Hahner

Scientist for Machine Learning

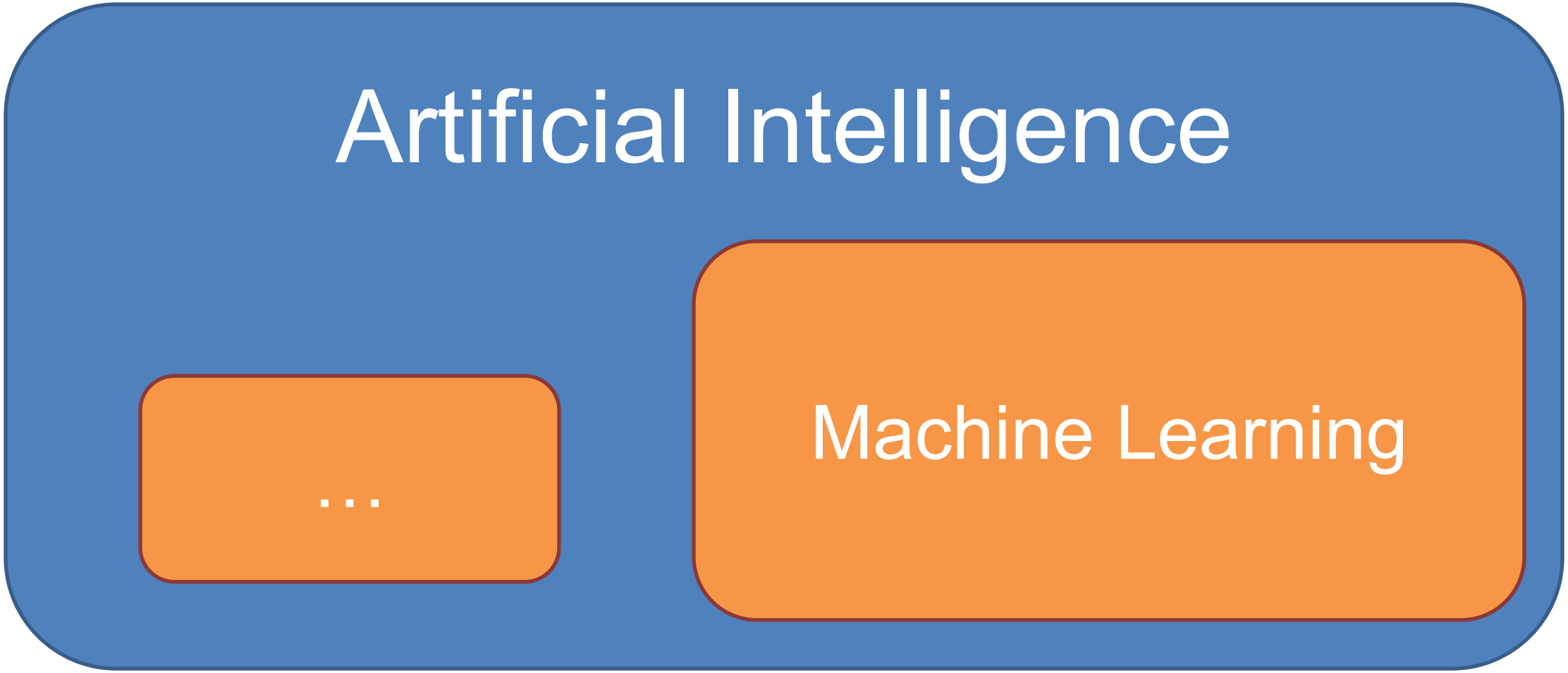
Based on slides from Jesper Dramsch

Outline

- Understanding AI & Machine Learning
- Types of Machine Learning
- Key Concepts in Machine Learning
- Dealing with Data
- Finding the Optimal Model

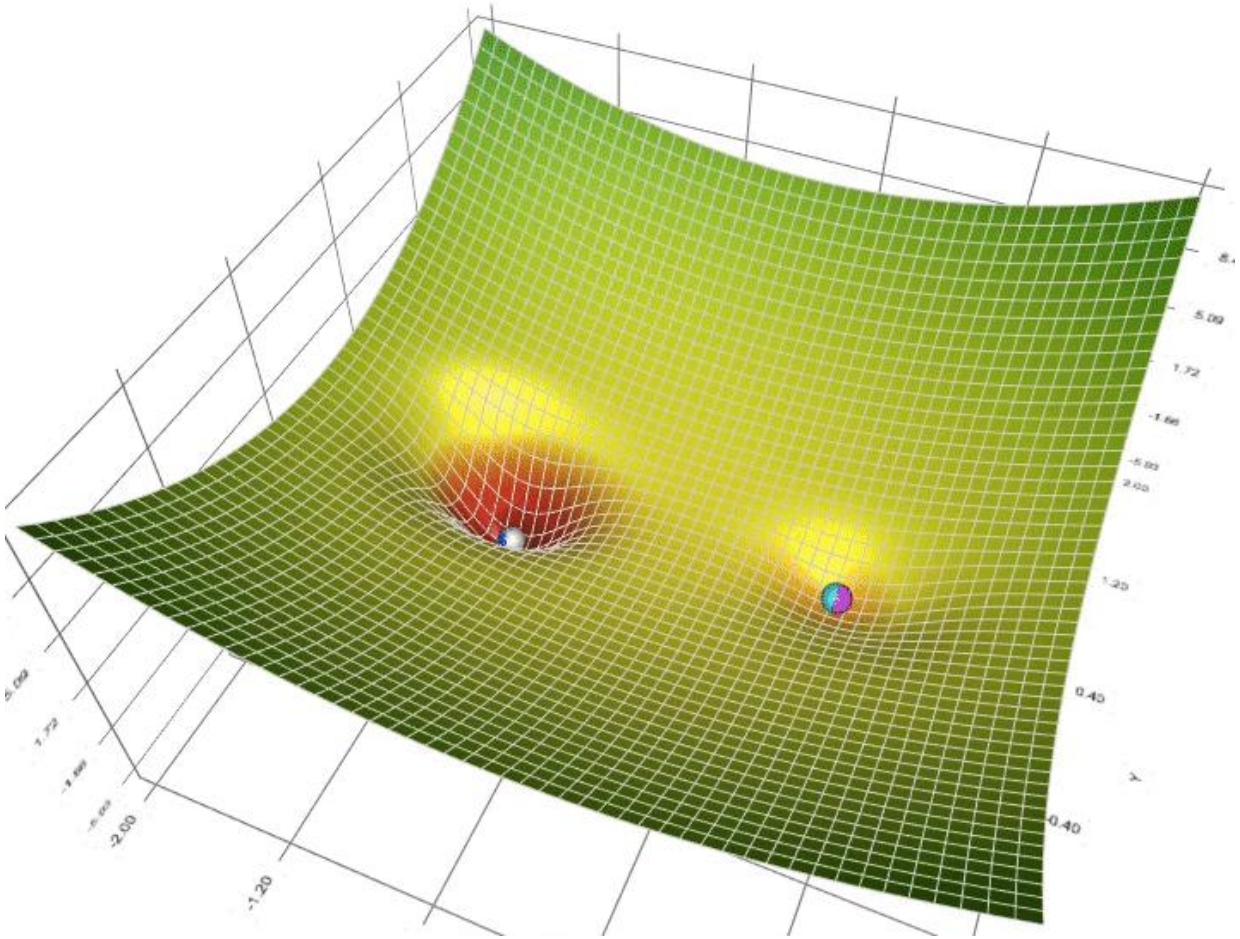
Understanding AI & ML

Machine Learning and Artificial Intelligence



“Machine Learning is a set of **algorithms** that improve their **performance** on a set **task** through **experience**.”

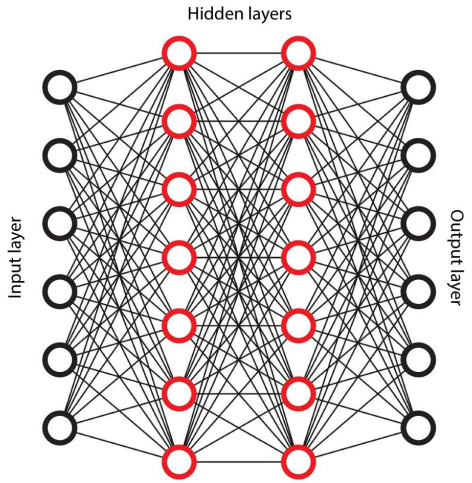
Statistical Methods & Numerical Optimisation



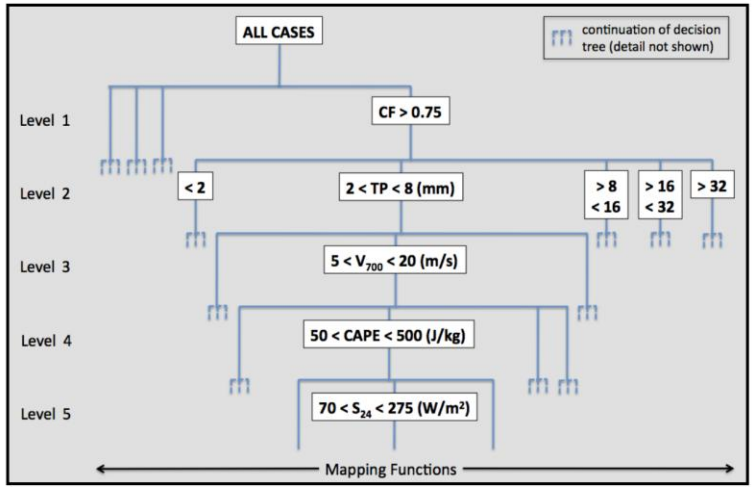
“Gradient Descent iteratively adjusts model parameters to minimise error, using the slope of the loss function to guide updates.”

Many Models – Many Methods

Neural Networks

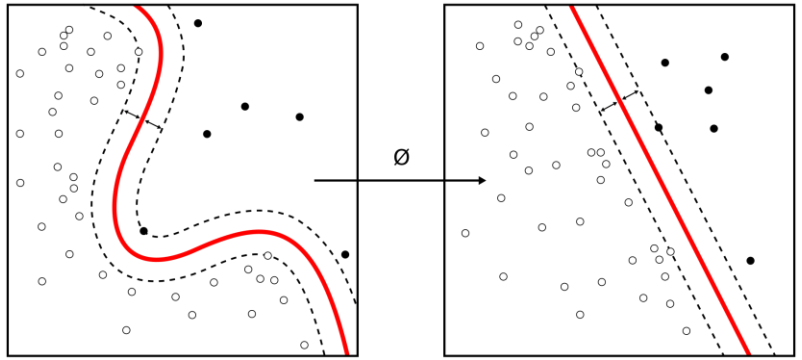


Decision Trees



Hewson and Pillosu 2020

Support-Vector Machine



14:00 Neural Networks and Deep Learning

12:00 Decision Trees and Random Forests

Why I'm keeping the details on models short

- Modern software packages make machine learning easy!

Choose a model

```
>>> from sklearn import svm
>>> clf = svm.SVC(gamma=0.001, C=100.)
```

Fit the model to training data

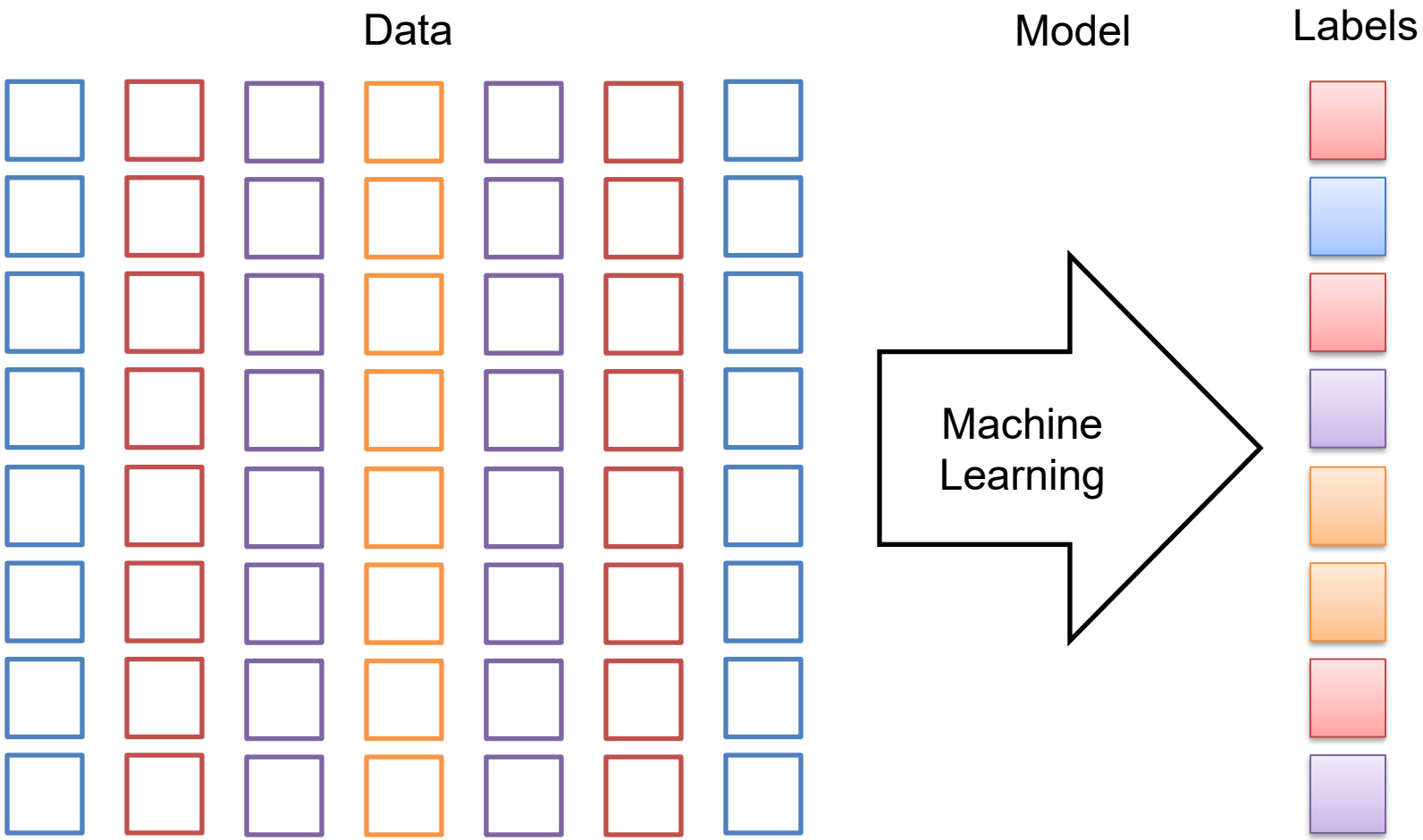
```
>>> clf.fit(digits.data[:-1], digits.target[:-1])
SVC(C=100.0, gamma=0.001)
```

Use model to predict

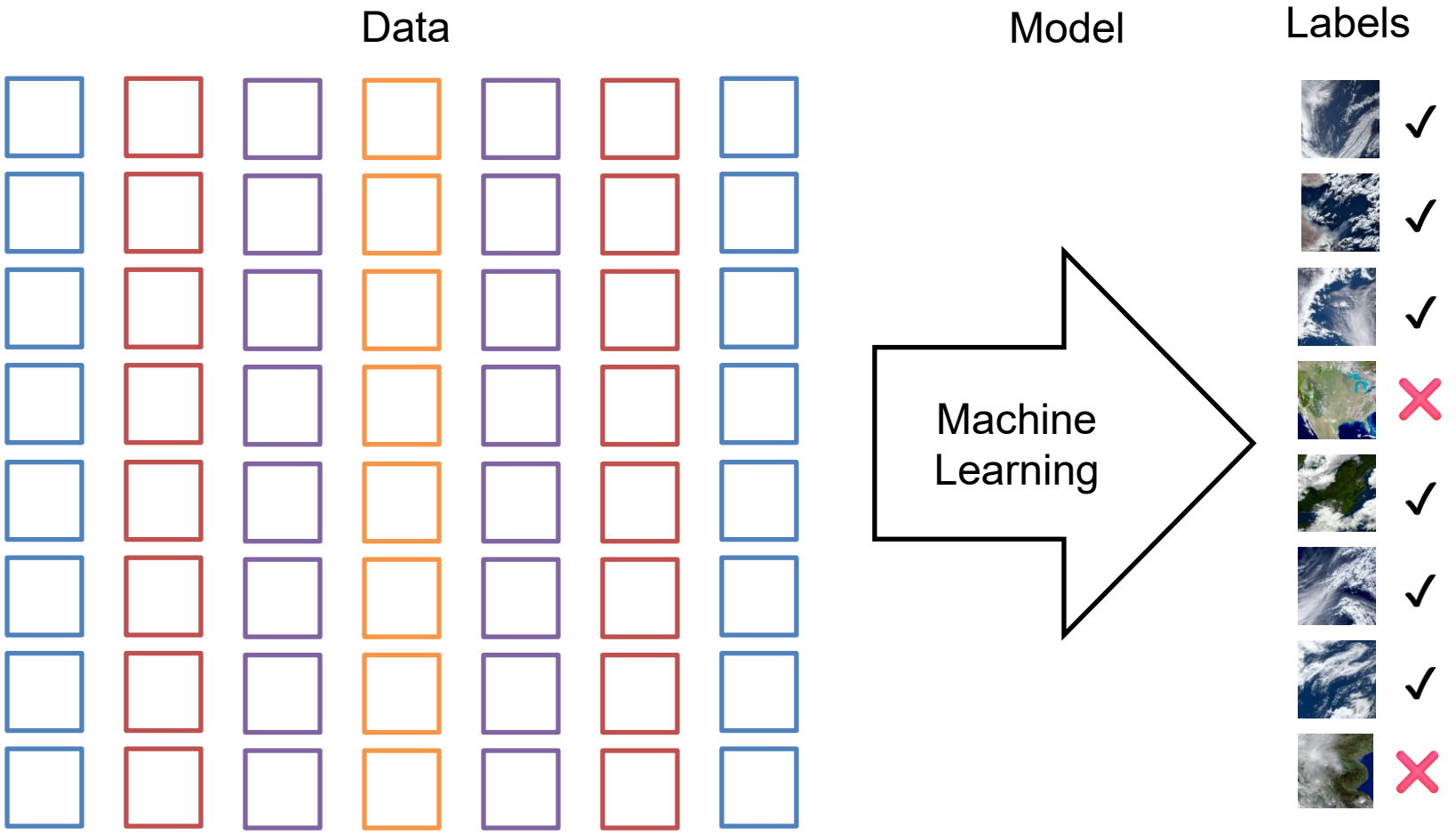
```
>>> clf.predict(digits.data[-1:])
array([8])
```

Types of Machine Learning

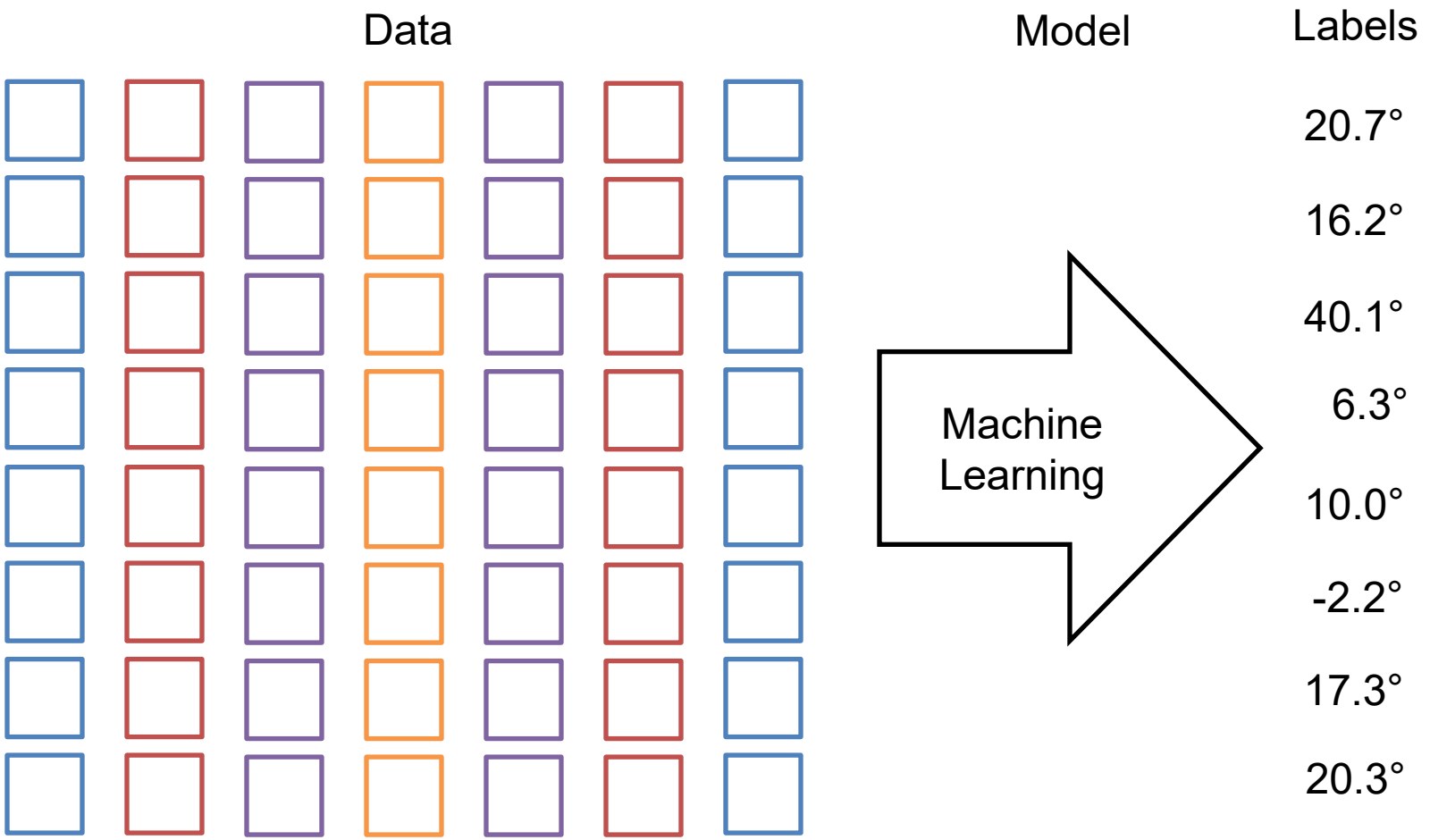
Supervised Learning



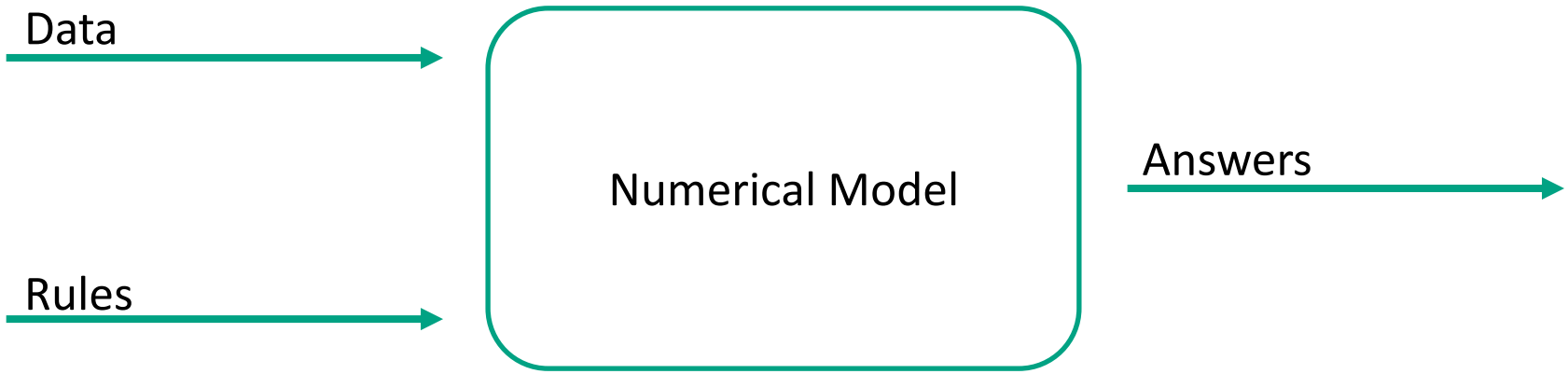
Supervised Learning – Classification



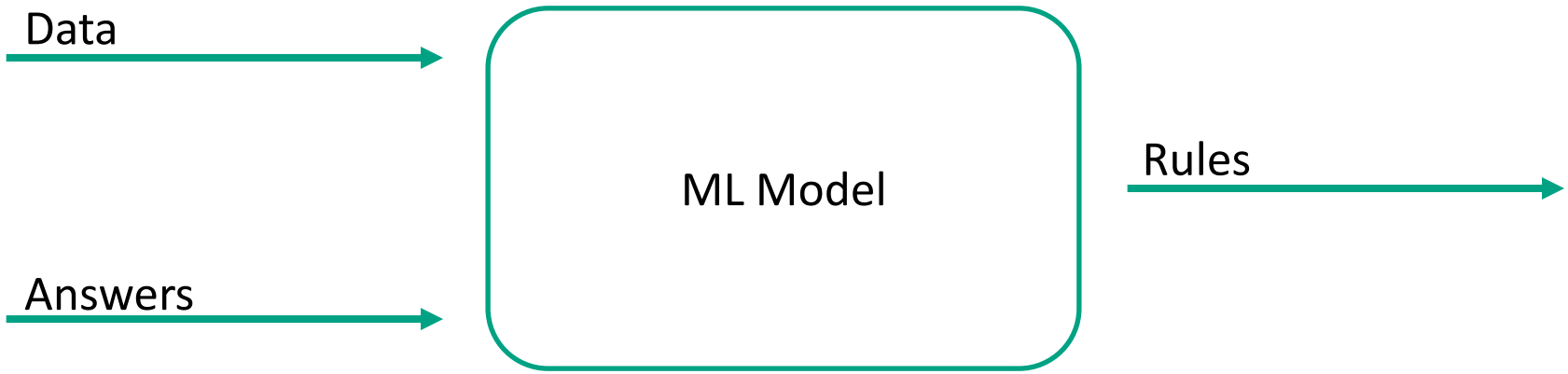
Supervised Learning – Regression



Classical Modelling

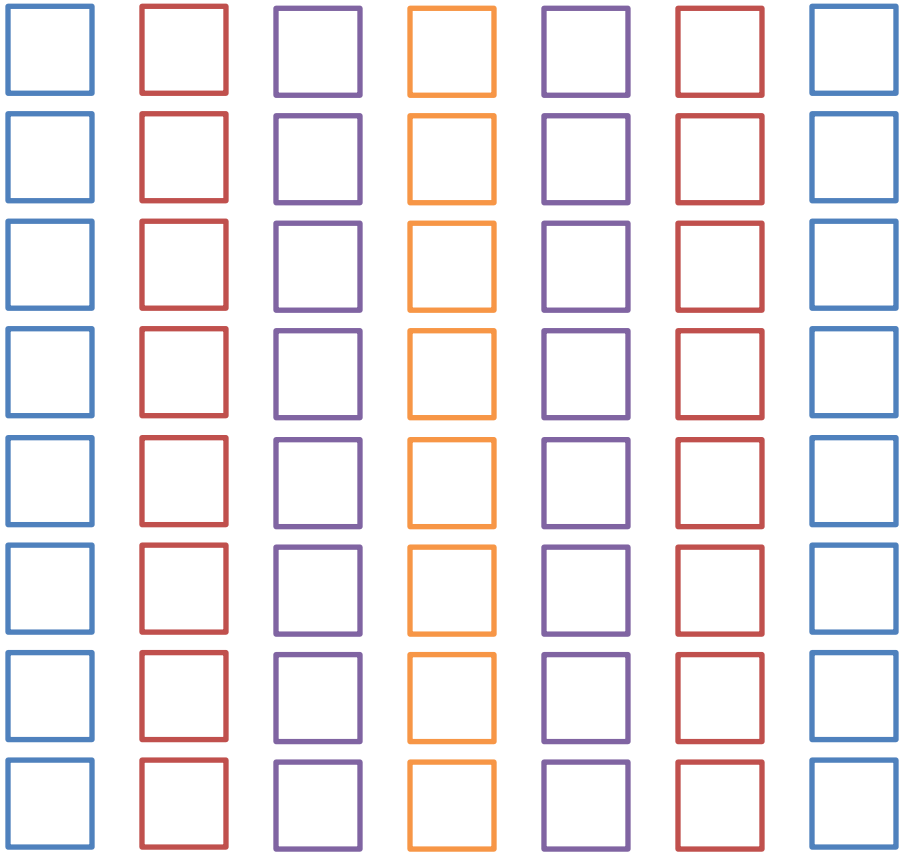


Supervised Machine Learning Modelling



Unsupervised Learning

Data



Labels

?

?

?

?

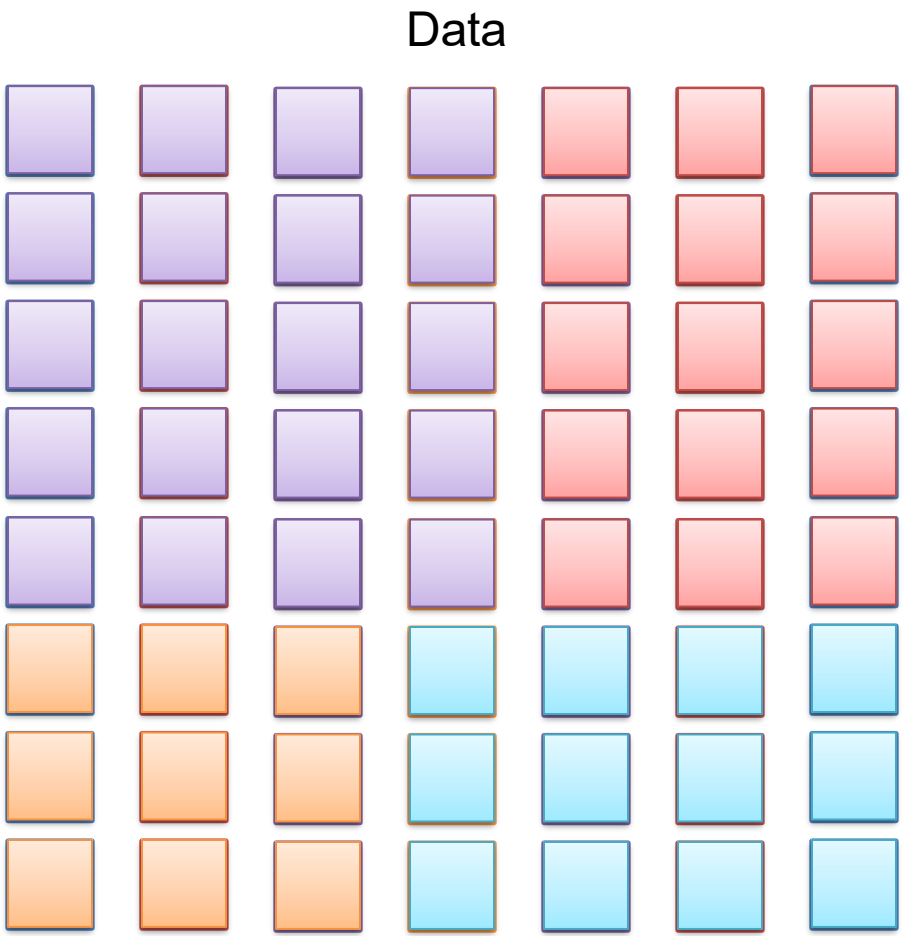
?

?

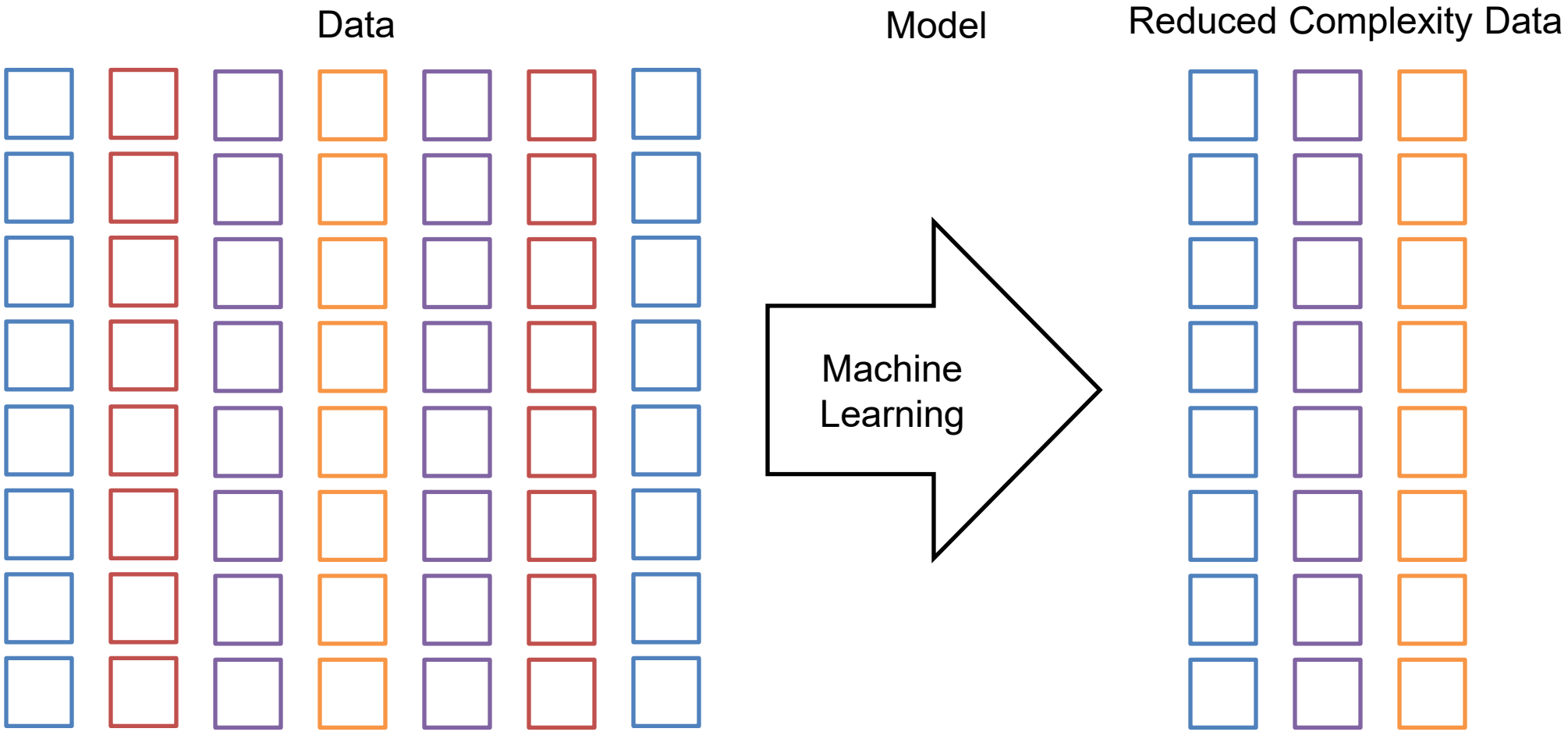
?

?

Unsupervised Learning – Clustering



Unsupervised Learning – Dimensionality Reduction

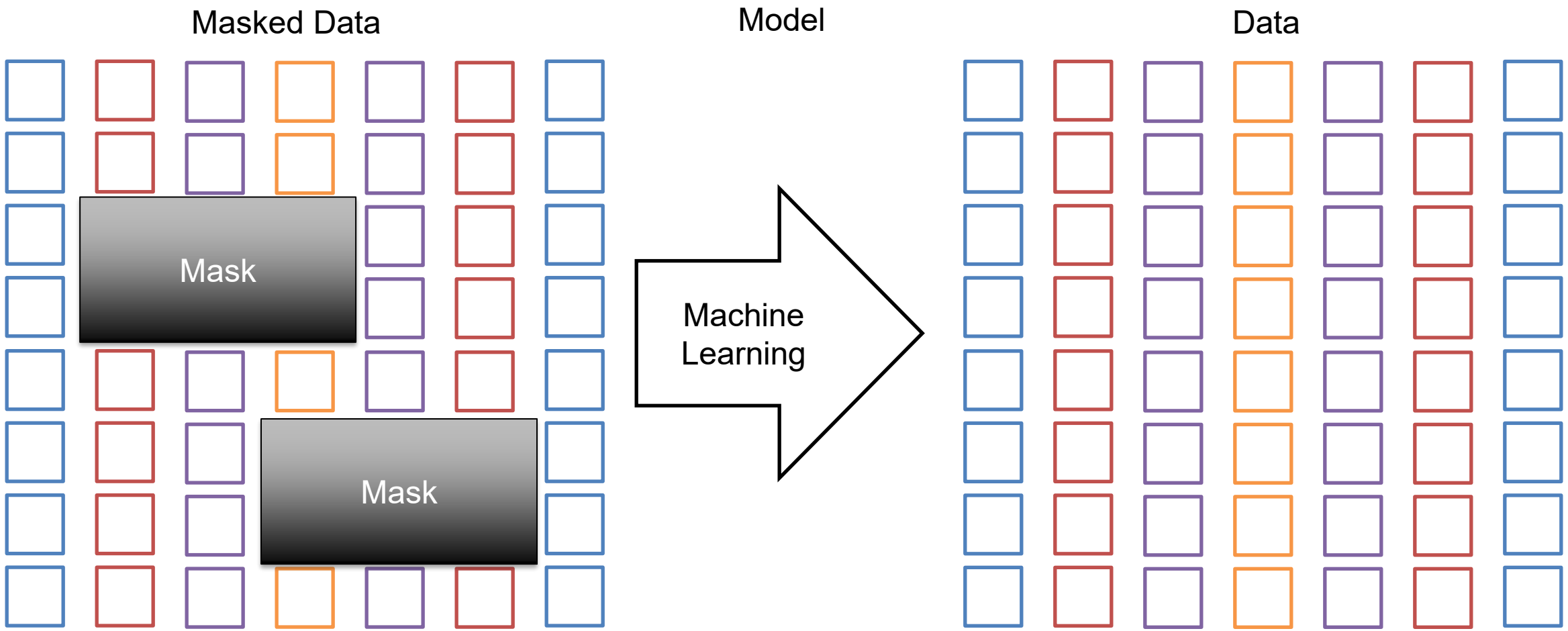


Unsupervised Machine Learning

- Unlabeled data
  - Labeling needs expertise and is expensive
  - Labeling can introduce bias
- Exploits the internal structure of data
- Can accomplish different tasks
  - Assign Labels
  - Reduce complexity of data

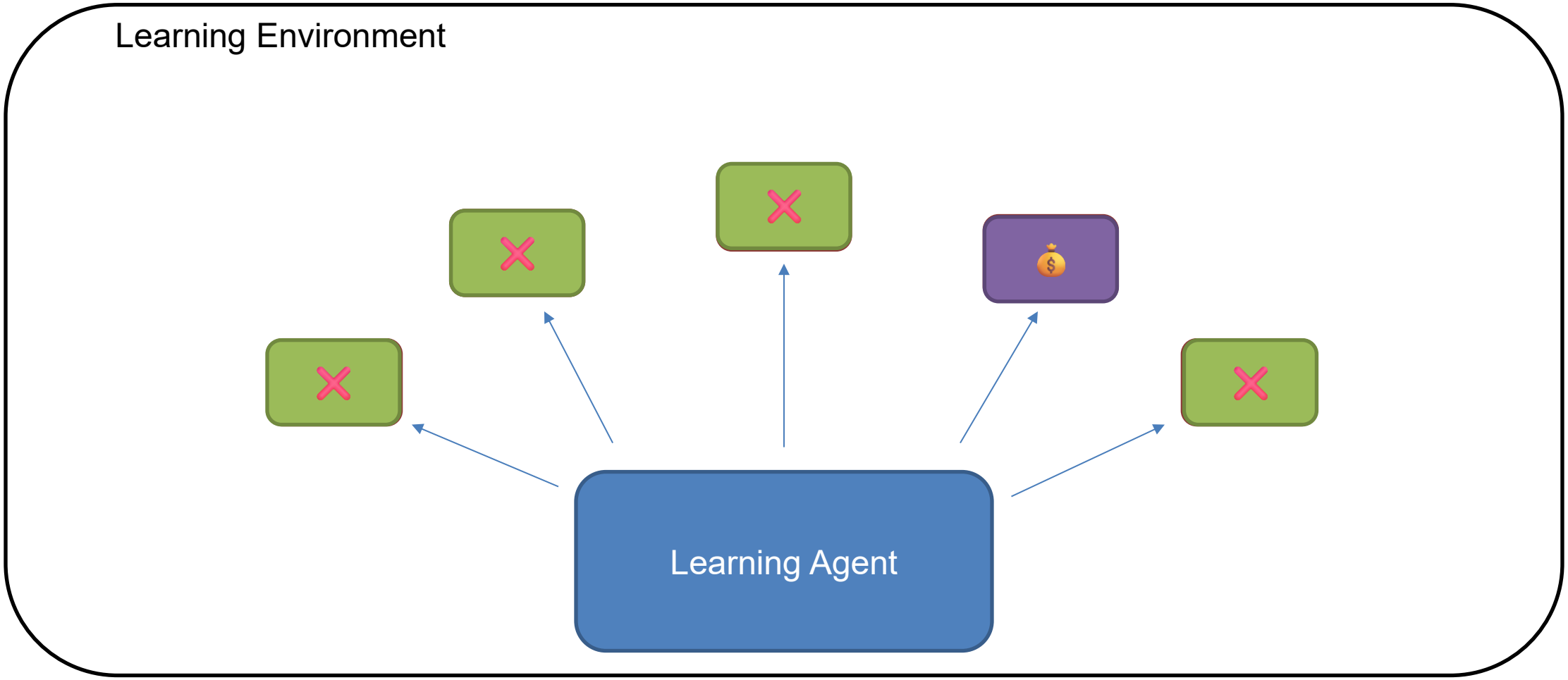
Self-supervised Learning

Wednesday at 10:45
Self-supervised Learning



ECMWF

Reinforcement Learning



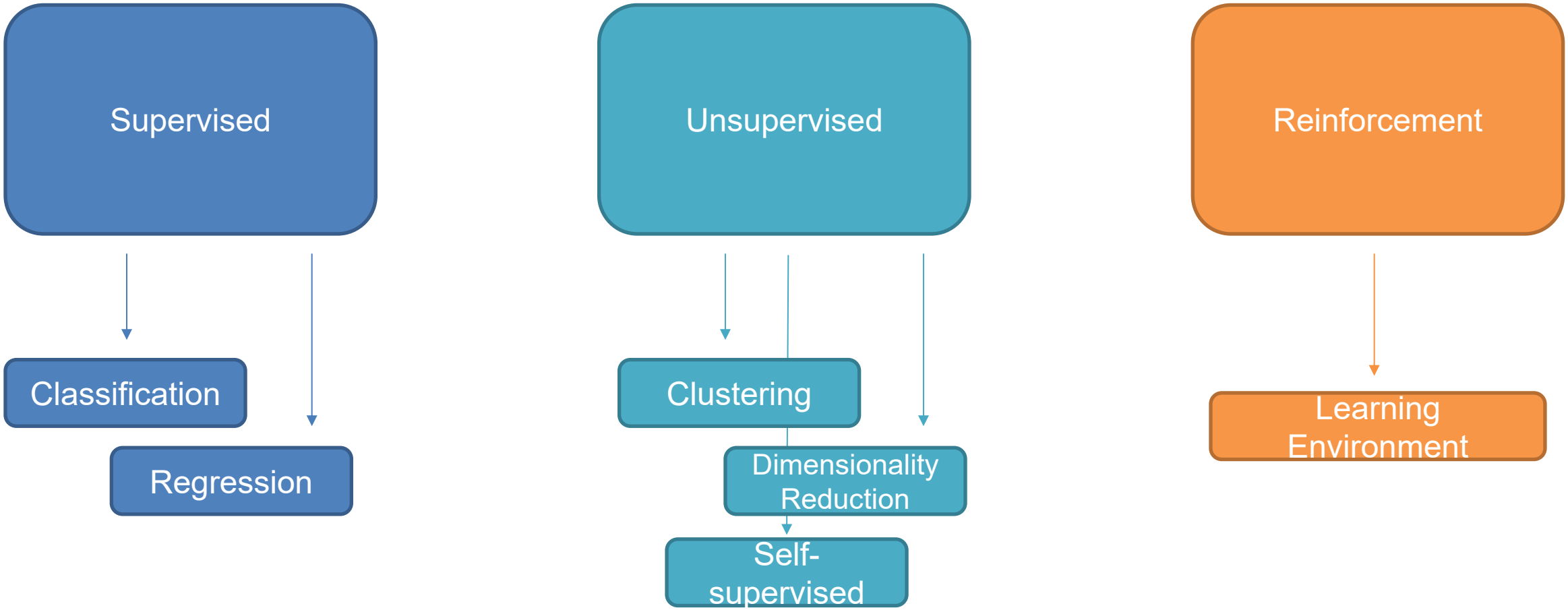
Reinforcement Learning – Games

ECMWF

Reinforcement Learning – Real World

ECMWF

The Types of Machine Learning



ECMWF

Other “Learning” which is not a “Type”

Deep learning and artificial neural networks as one example of machine learning

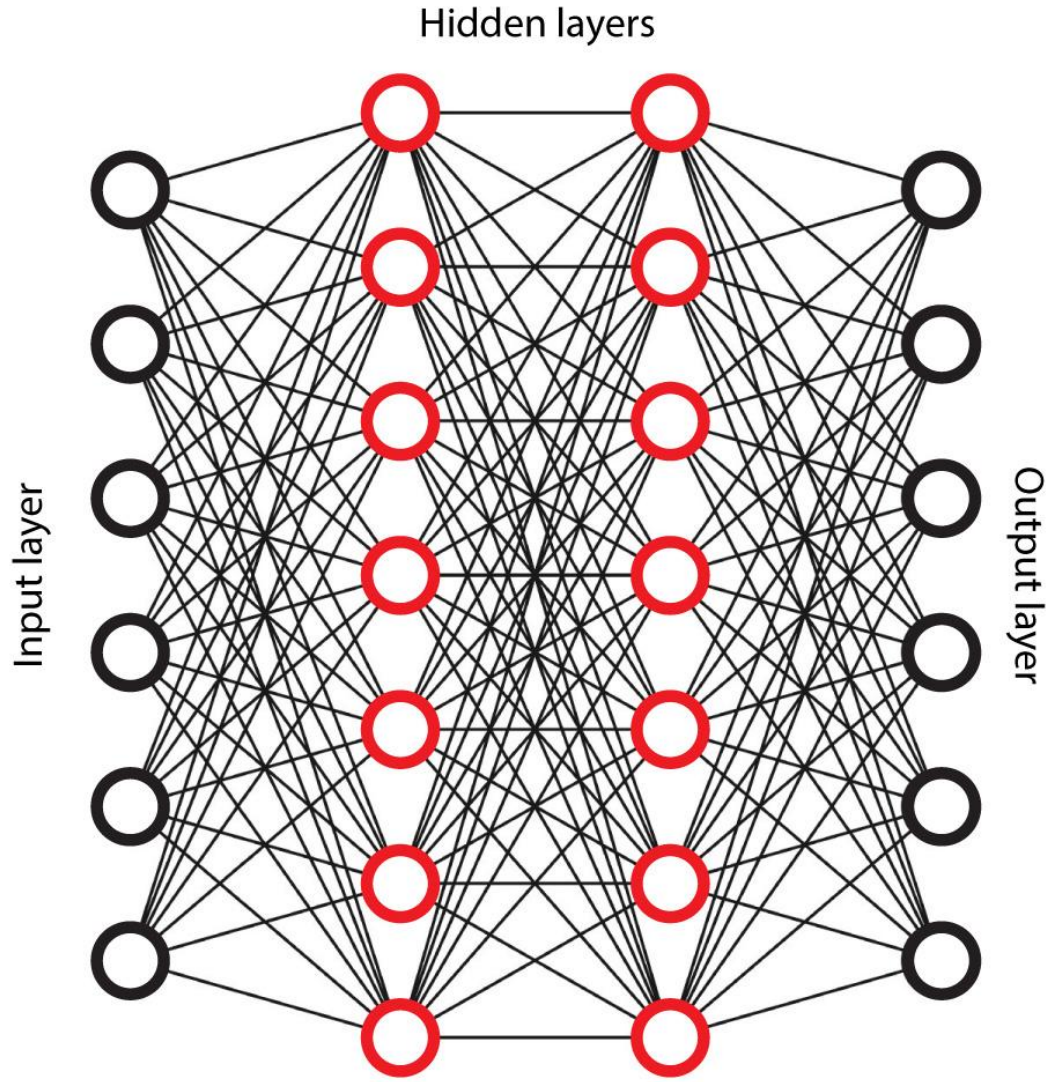
The concept:
Take input and output samples from a large data set
Learn to predict outputs from inputs
Predict the output for unseen inputs

The key:
Neural networks can learn a complex task as a “black box”
No previous knowledge about the system is required
More data will allow for better networks

The number of applications is increasing by day:
Image recognition
Speech recognition
Healthcare
Gaming
Finance
Music composition and art
…

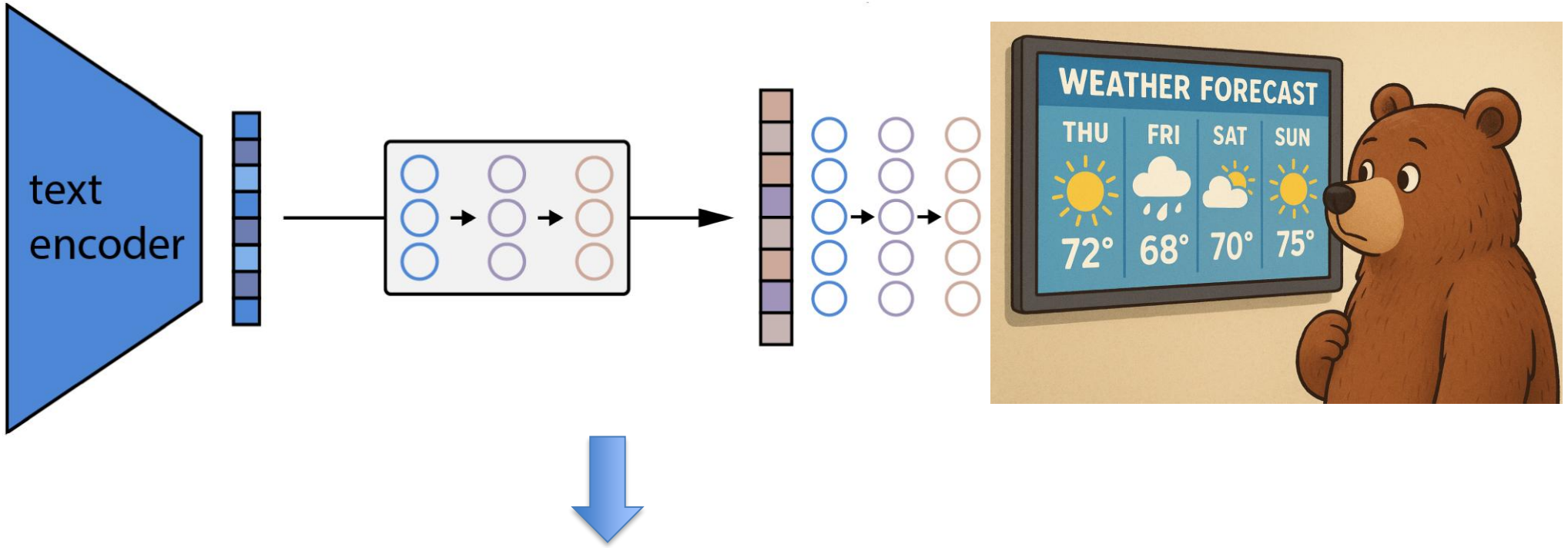
And weather/climate!

14:00 Neural Networks and Deep Learning



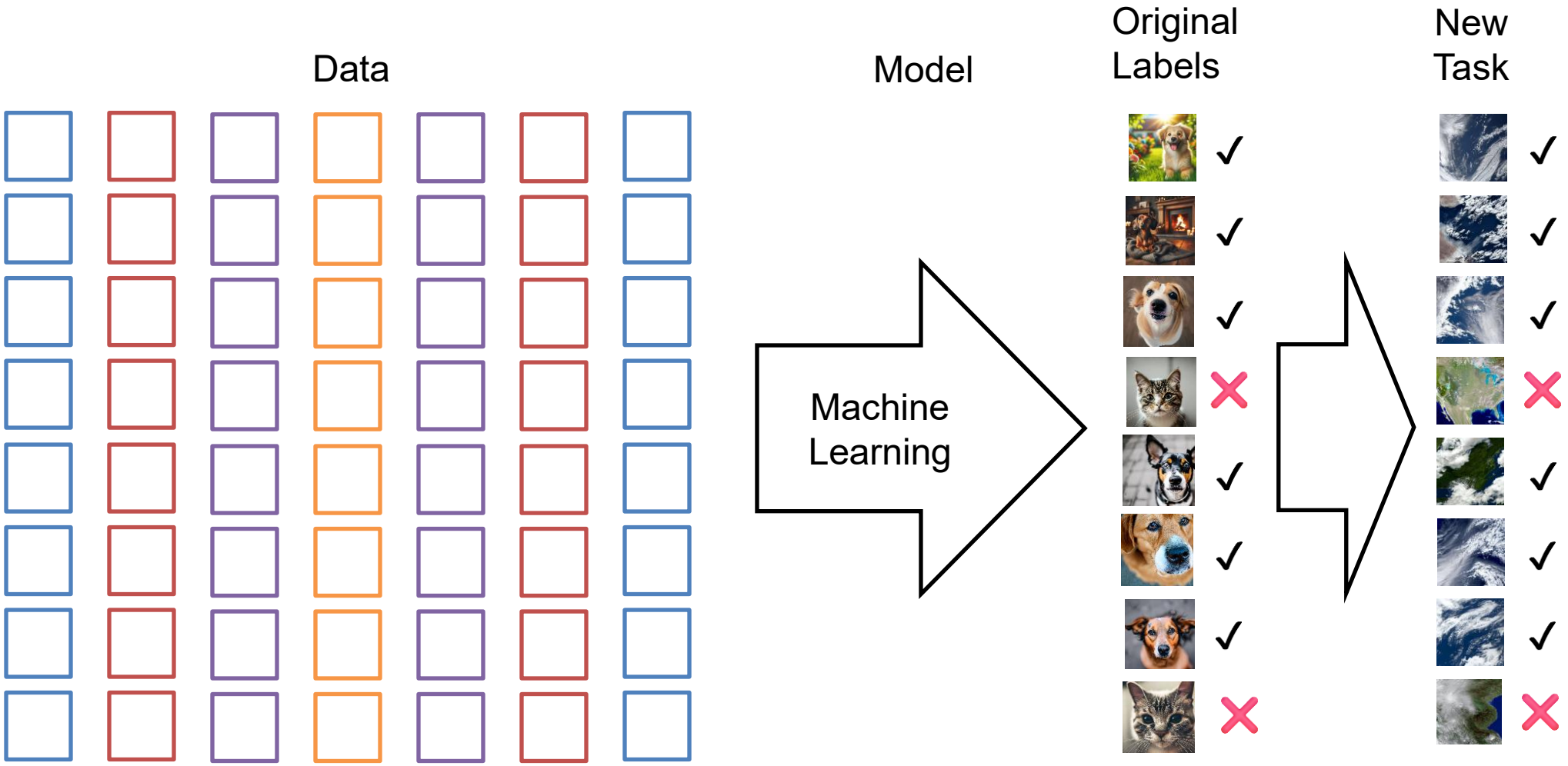
Deep Generative Models

A bear looking at a weather forecast



Wednesday 9:00
Deep Generative Models

Transfer Learning and Domain Adaptation



ECMWF

Key Concepts in Evaluation of Machine Learned Models

Training, Validation and Testing

Disjoint Datasets	Purpose
Training Set	Fit the model (learn parameters)
Validation Set	Tune model architectures and select best model
Test Set	Final unbiased evaluation after training

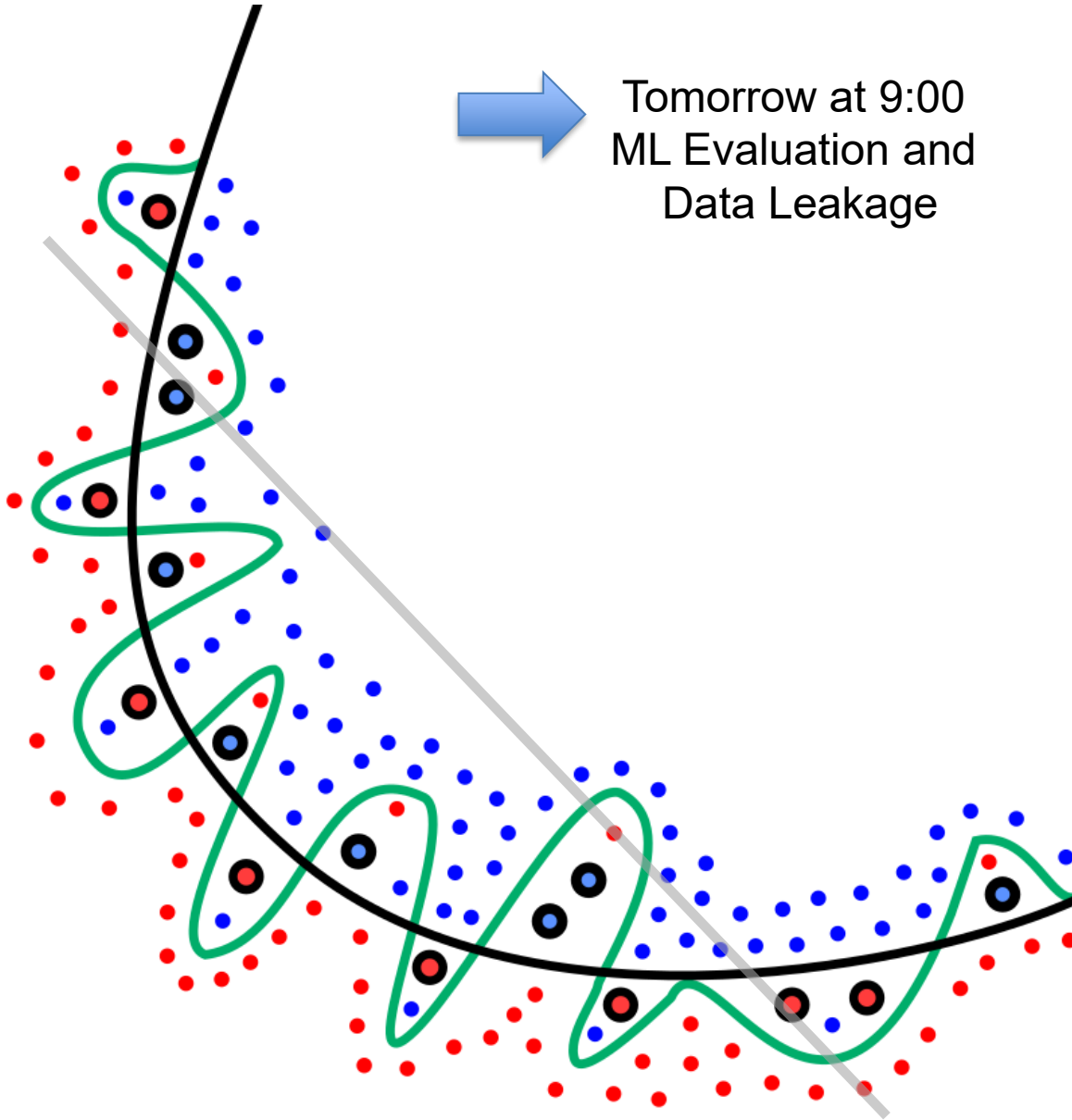
- Workflow

1. Train the model on training data
2. Validation using validation set -> adjust model
3. Test the final model on test data -> estimate true performance

“Generalization is a ML **model’s ability** to generate accurate and reliable predictions on previously **unseen data**.”

Generalization and Overfitting

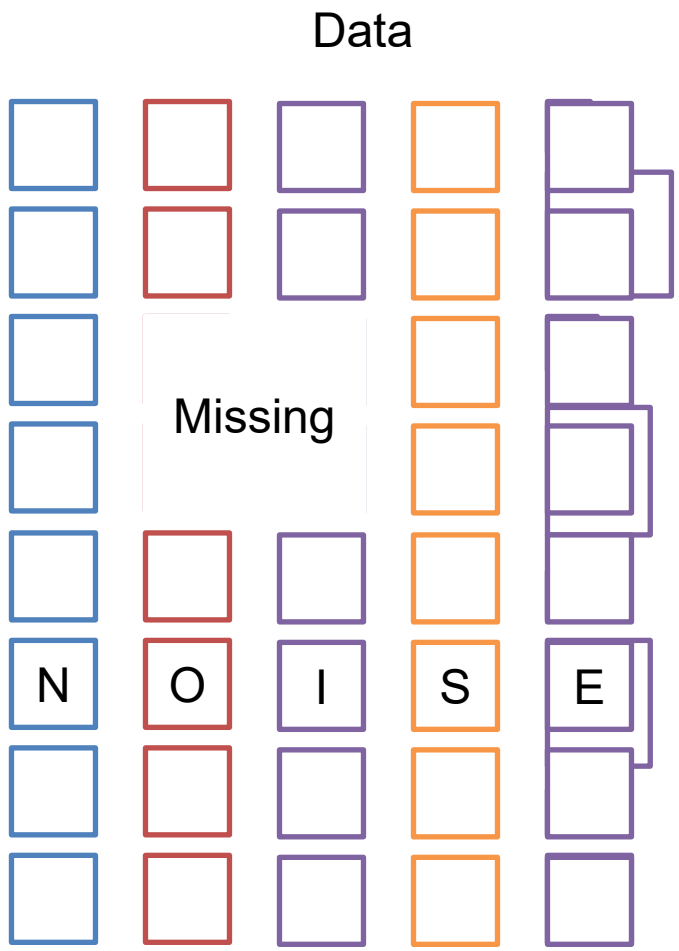
- ML model learns from historic data
- Generalization for performance on unseen data
- Underfitting
  - Model can’t fit the complex data
- Overfitting
  - Model exactly fits the training data
  - Does not generalize to unseen data
- Overfitting can be avoided by
  - Reducing model complexity
  - Regularization
  - Pruning
  - Etc.



Tomorrow at 9:00
ML Evaluation and
Data Leakage

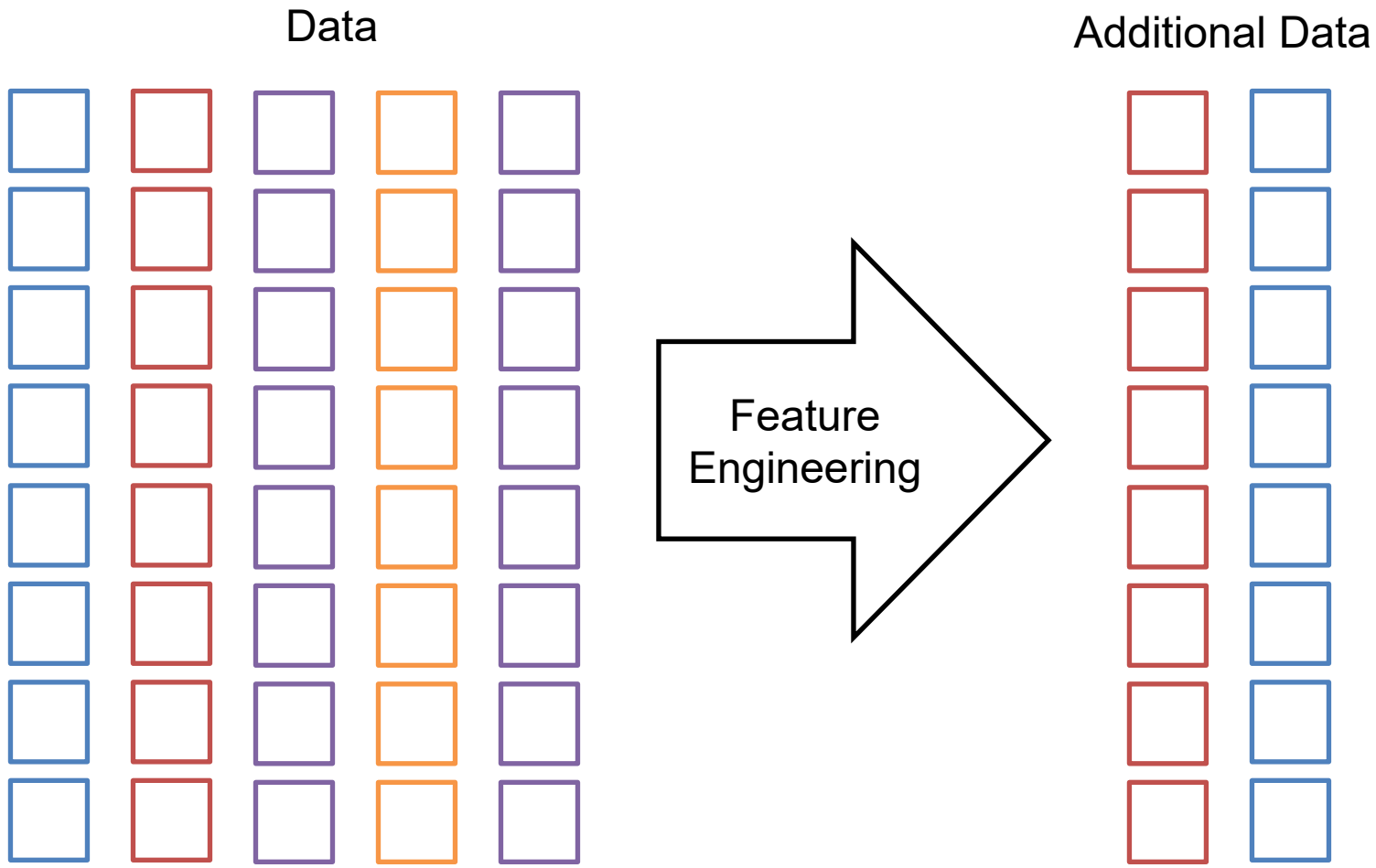
Dealing with Data

Data Preprocessing



- Machine learning models struggle with irregular data
- Imputation
  - Filling in missing values
  - Often with Mean or Median
- Data Cleaning
  - Removing noise from data
  - Careful! Easy to ”over-clean”
  - Needs to be faithful to real-world data
- Normalisation
  - Standardization
  - Min-Max Scaling
- Transformations
  - Log-Scaling

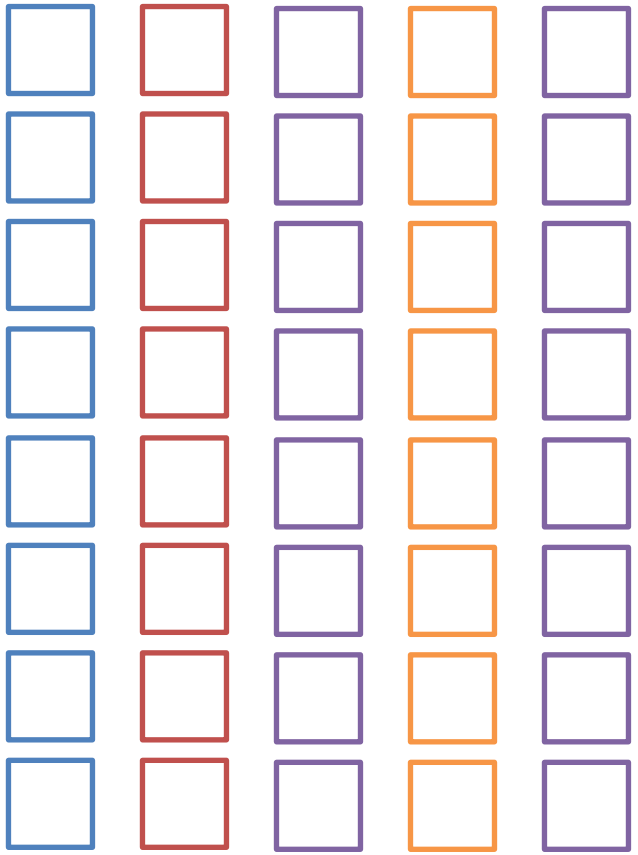
Feature Engineering



Managing Big Data with Batch Processing

Tomorrow at 11:00
Data Handling and Infrastructure

Big Data



Batch 1

Batch 2

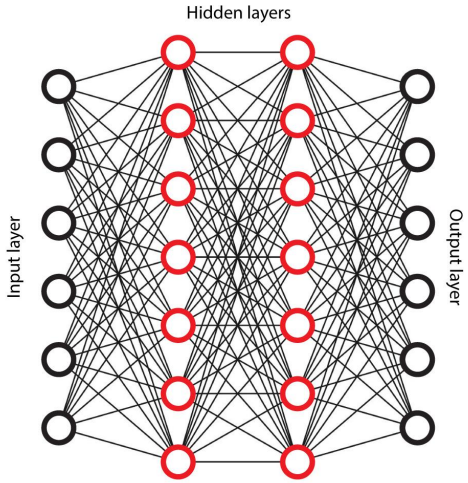
Hyperparameters and finding the optimal model

“There is no **Free Lunch**.”

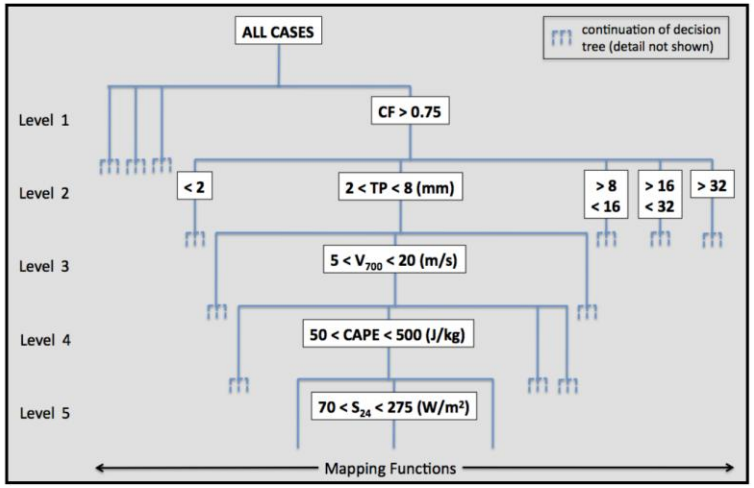
Hyperparameters and Tuning

- Parameters:
  - Learned from data during training
- Hyperparameters:
  - ”Settings of Model”
- Examples of Hyperparameters:
  - Number of nodes
  - Number of layers
  - Number of Trees
  - Learning Rate of optimization process
  - Batch size, of incremental training

Neural Networks

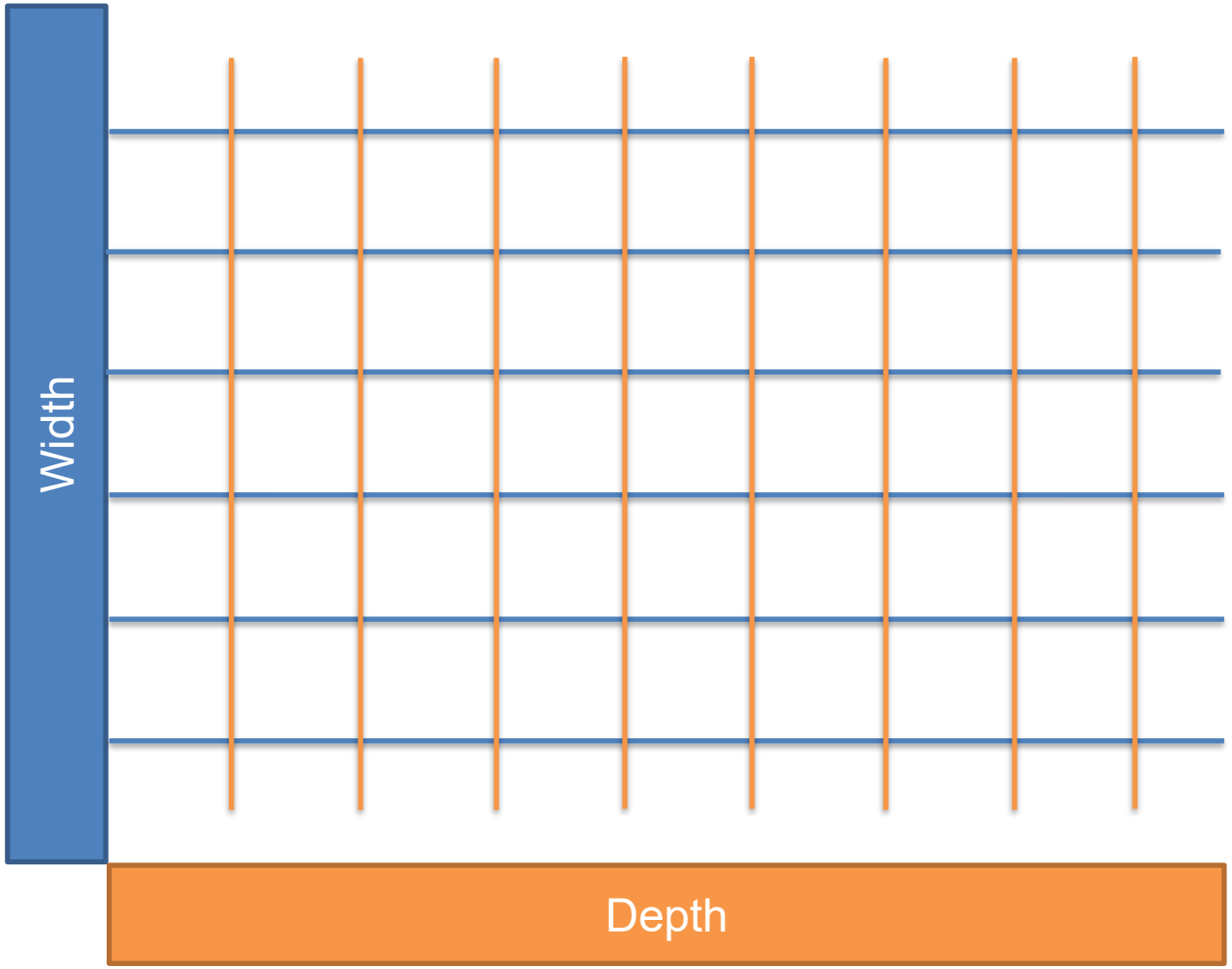


Decision Trees



Hewson and Pillosu 2020

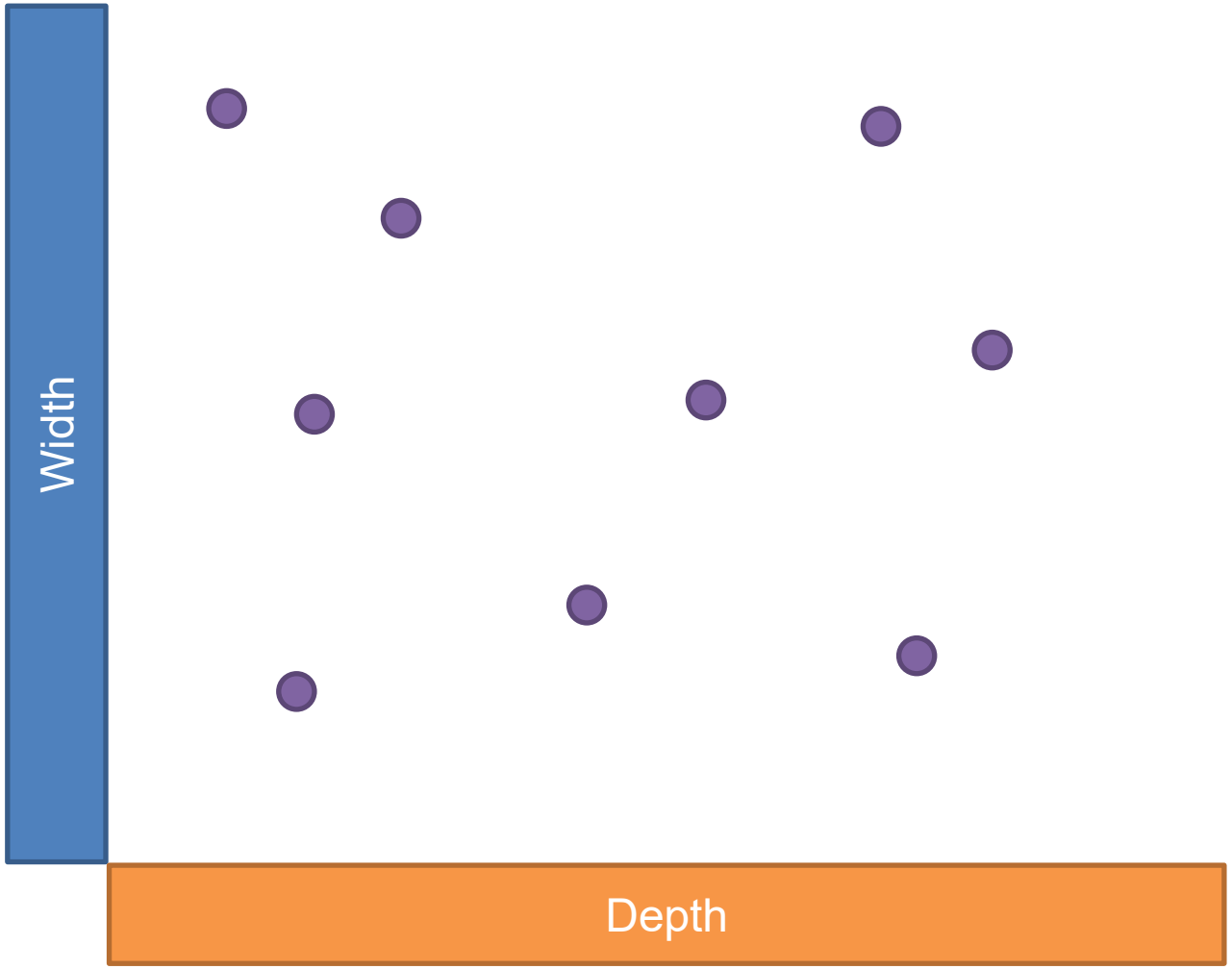
Grid search



- Exhaustive Search
- Every Combination is Evaluated
- Combinatoric Explosion of Evals
- Inefficient searching beyond minimum
- Possible to miss optimal parameters because explicit values are provided

```
1 from sklearn.model_selection import GridSearchCV
2
3 parameters = {'width':[5, 10, 15, 20],
4               'depth':[1, 2, 3, 4, 5, 6],
5               'activation':['tanh', 'relu']}
6
7 gridcv = GridSearchCV(neural_network, parameters)
8
9 gridcv.fit(X_train, y_train)
```

Randomized search



- Exhaustive Search
- Budget independent of No. parameters
- Adding Parameters not Inefficient
- Inefficient searching beyond minimum
- Possible to miss optimal parameters

because explicit values are provided

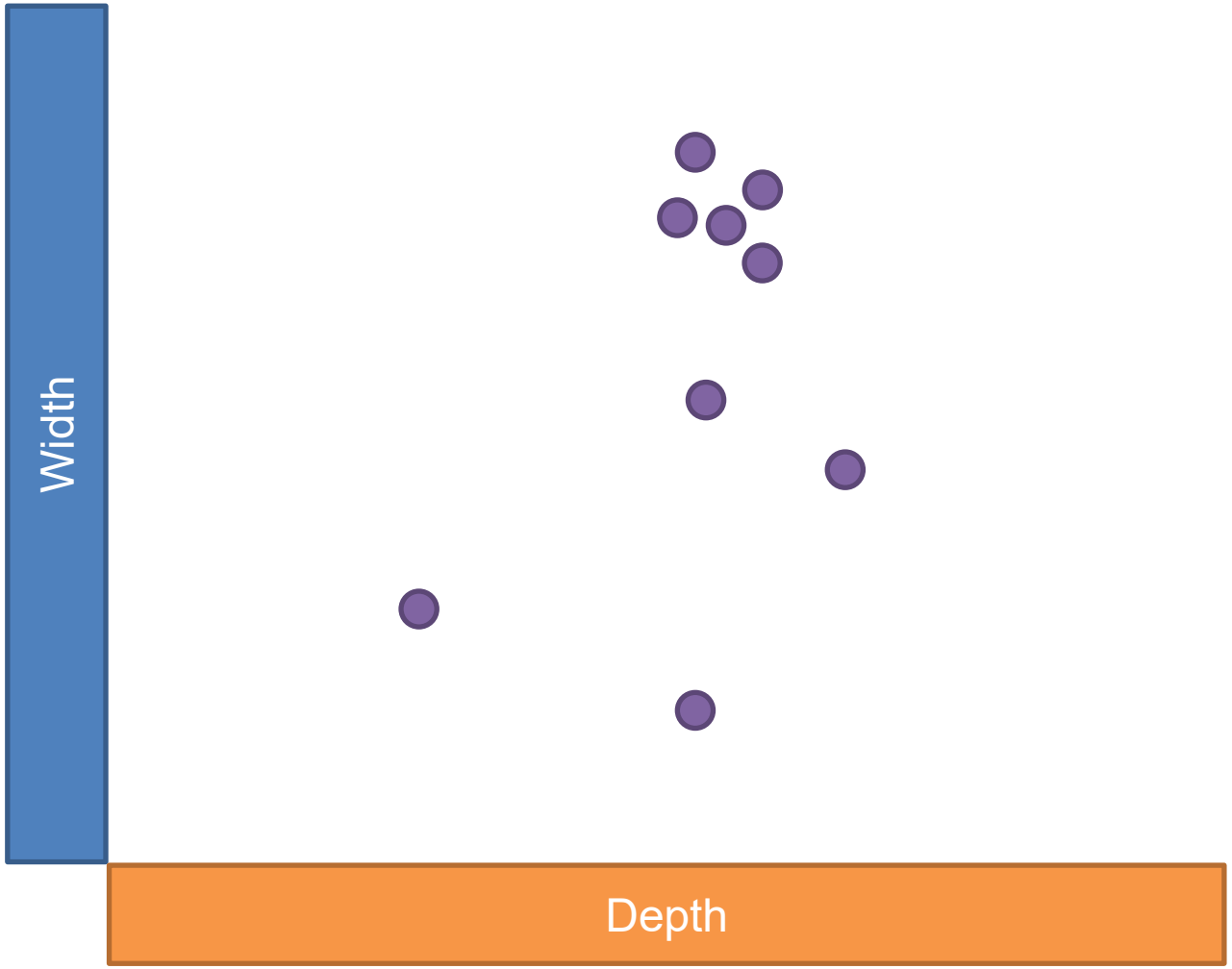
```
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import uniform

distributions = {'width': uniform(5, 15),
                 'depth': uniform(1, 5),
                'activation':['tanh', 'relu']}

randomcv = RandomizedSearchCV(neural_network, distributions)

randomcv.fit(X_train, y_train)
```

Bayesian search



- Search based on former parameters
- Bayesian Optimization
- Converges to a minimum
- Adding Parameters adds complexity
- Unimportant parameters complicate optimization significantly

```
from skopt import BayesSearchCV

distributions = {'width': (5, 20, 'uniform'),
                 'depth': (1, 6, 'uniform'),
                 'activation':['tanh', 'relu']}

randomcv = BayesSearchCV(neural_network, distributions)

randomcv.fit(X_train, y_train)
```

Conclusion

What we Learned

- AI and Machine Learning are related but distinct
- Open-source software makes ML easier
- Types of machine learning model:
  - Un- and Supervised learning
  - Reinforcement Learning
- Other relevant “Learning”
  - Deep Learning
  - Transfer Learning
- Generalization and Overfitting
- Data-Preprocessing
- Hyperparameter tuning