Big Data Computing

Master's Degree in Computer Science

2022-2023

Gabriele Tolomei

Department of Computer Science

Sapienza Università di Roma

tolomei@di.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

Recap from Last Lecture(s)

2 unsupervised learning techniques to extract "structural" patterns from raw data

Recap from Last Lecture(s)

2 unsupervised learning techniques to extract "structural" patterns from raw data

Clustering

- Group together similar objects according to a specific distance function
- Formalized as an NP-hard optimization problem
- K-means and its variants as effective heuristics that work in practice

Recap from Last Lecture(s)

2 unsupervised learning techniques to extract "structural" patterns from raw data

Clustering

- Group together similar objects according to a specific distance function
- Formalized as an NP-hard optimization problem
- K-means and its variants as effective heuristics that work in practice

Principal Component Analysis (PCA)

- Reduce data dimensionality
- Automatically extract features from raw data
- Resort to computing the eigenvectors and eigenvalues of the covariance matrix

SUPERVISED LEARNING

Human vs. Computer

- Computers are designed to be programmed by humans in order to solve a task/problem quicker and better than humans

Human vs. Computer

- Computers are designed to be **programmed** by humans in order to solve a task/problem quicker and better than humans
- Example
  - Task/Problem: Find the maximum element of a list of 1 million unsorted numbers

Human vs. Computer

- Computers are designed to be **programmed** by humans in order to solve a task/problem quicker and better than humans
- Example
  - Task/Problem: Find the maximum element of a list of 1 million unsorted numbers
  - Solution/Algorithm: Scan all the numbers in the set and keep track of the largest found "so far"

Human vs. Computer

- Computers are designed to be programmed by humans in order to solve a task/problem quicker and better than humans
- Example
  - Task/Problem: Find the maximum element of a list of 1 million unsorted numbers
  - Solution/Algorithm: Scan all the numbers in the set and keep track of the largest found "so far"
  - Code/Program: Encode the algorithm above into one specific programming language (e.g., C/C++, Java, Python)

Programming a Computer

Problem

Programming a Computer

Problem

Solution/Algorithm
explicitly designed by human

Programming a Computer

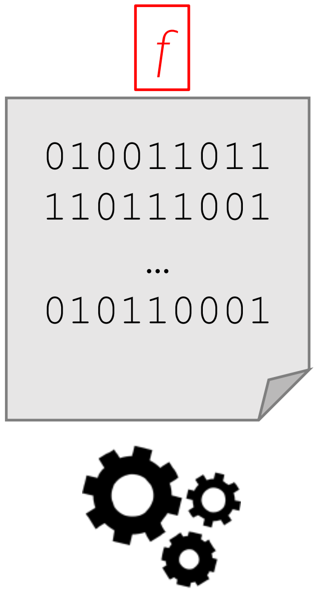
Problem

Solution/Algorithm
explicitly designed by human

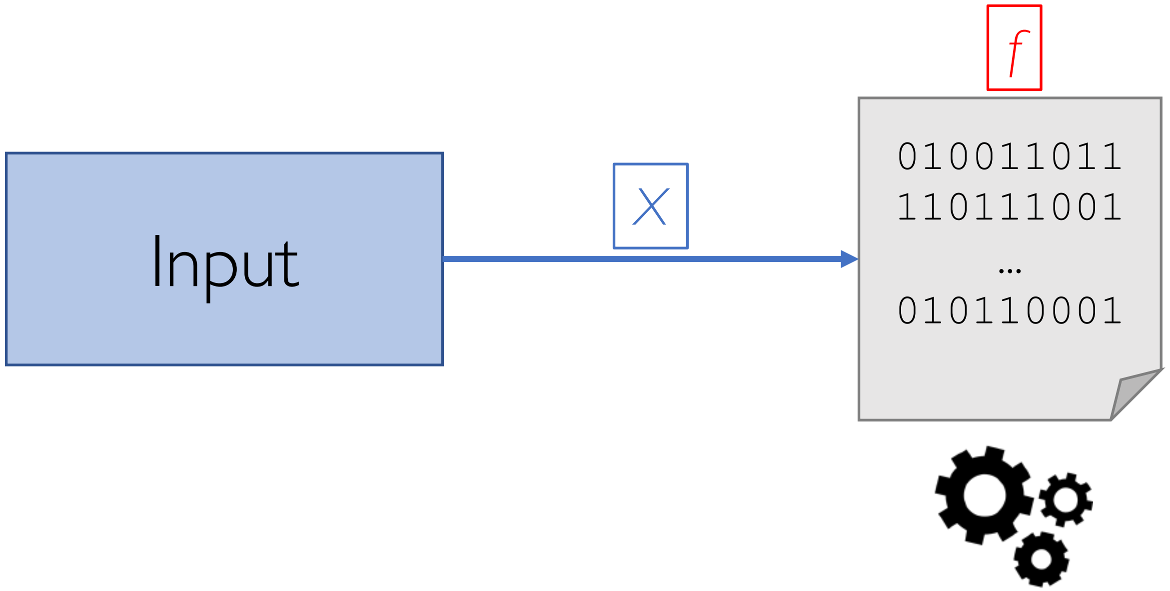
```
010011011
110111001
…
010110001
```

Encoding of
Computable Function f

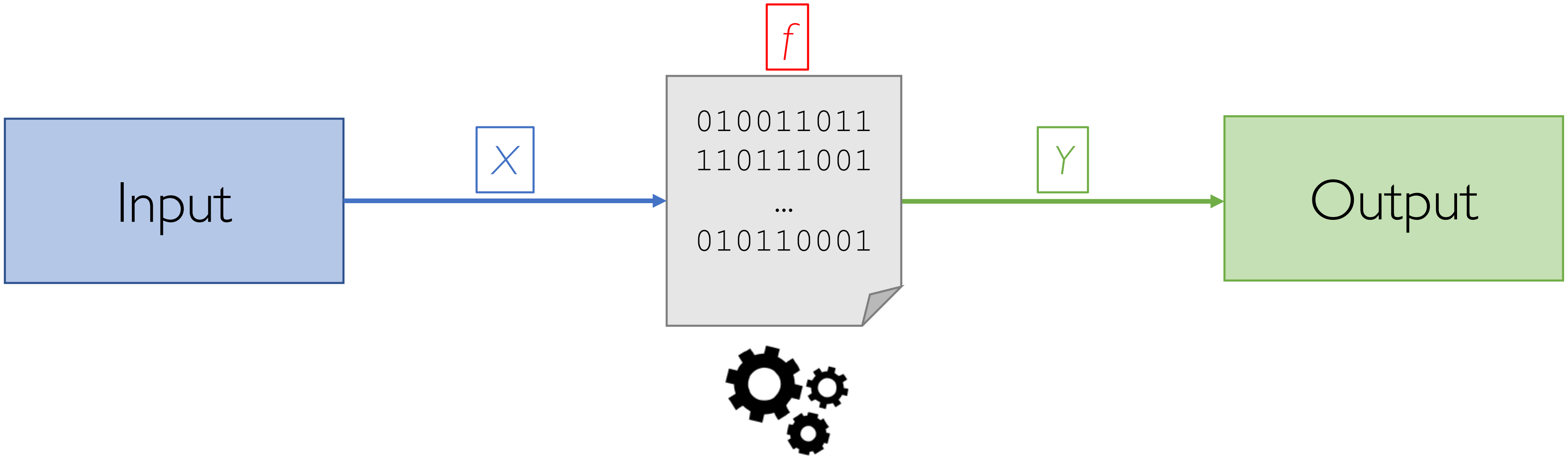
Programming a Computer



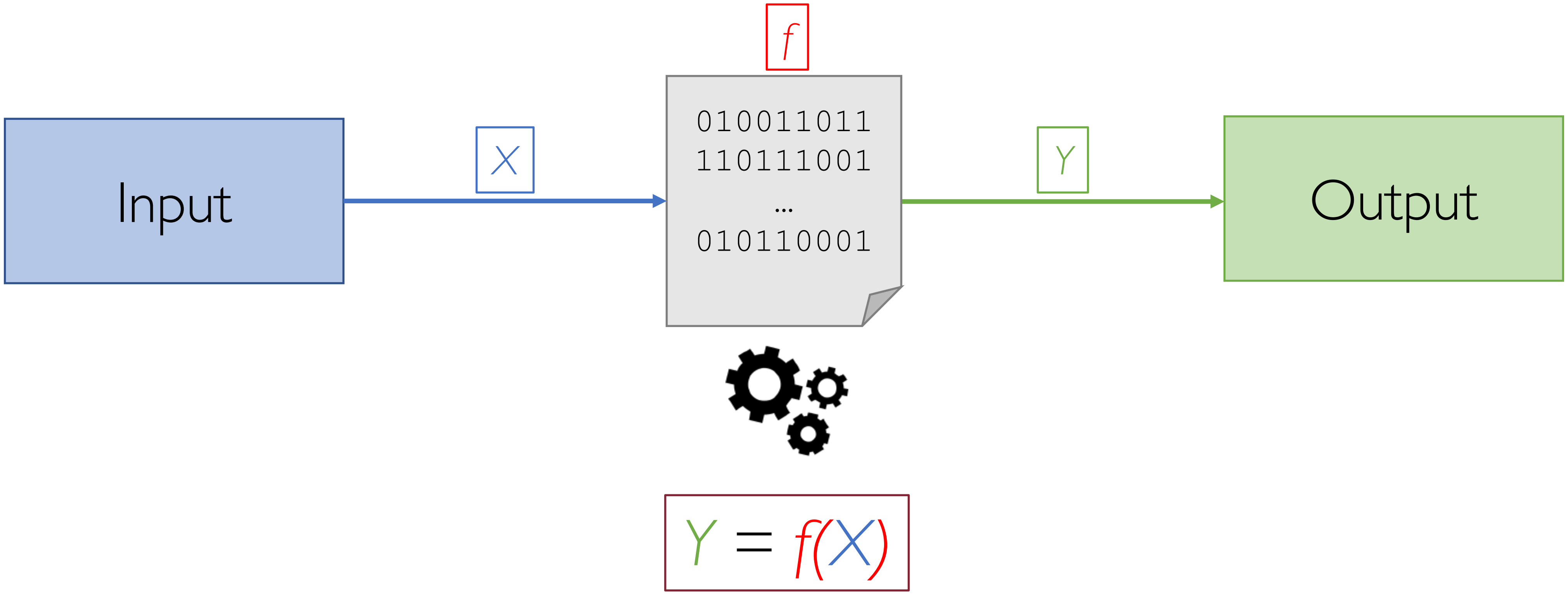
Programming a Computer



Programming a Computer



Programming a Computer



Can We Always Do That?

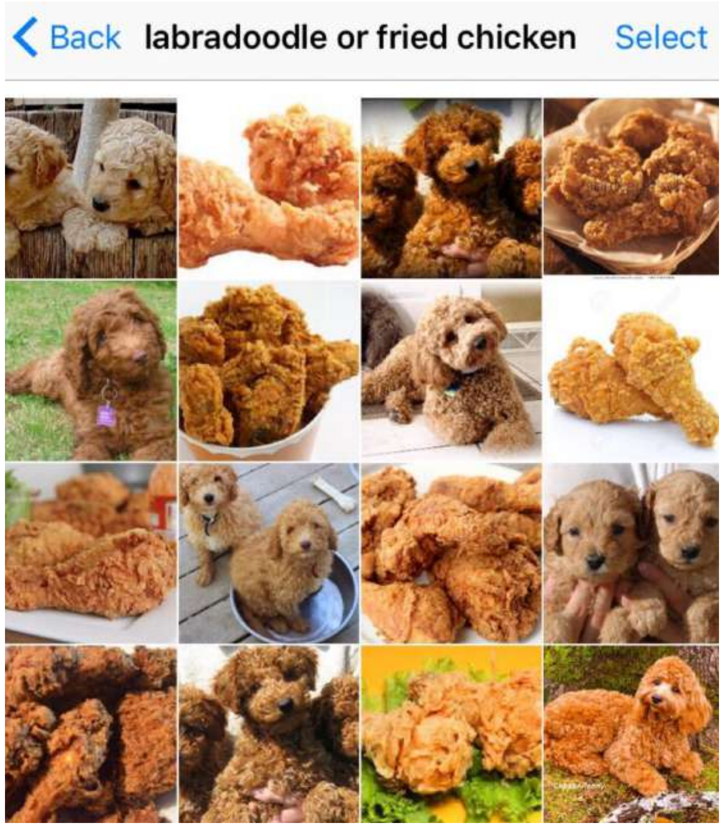
Chihuahua or Muffin?

[Copyright @teenybiscuit]

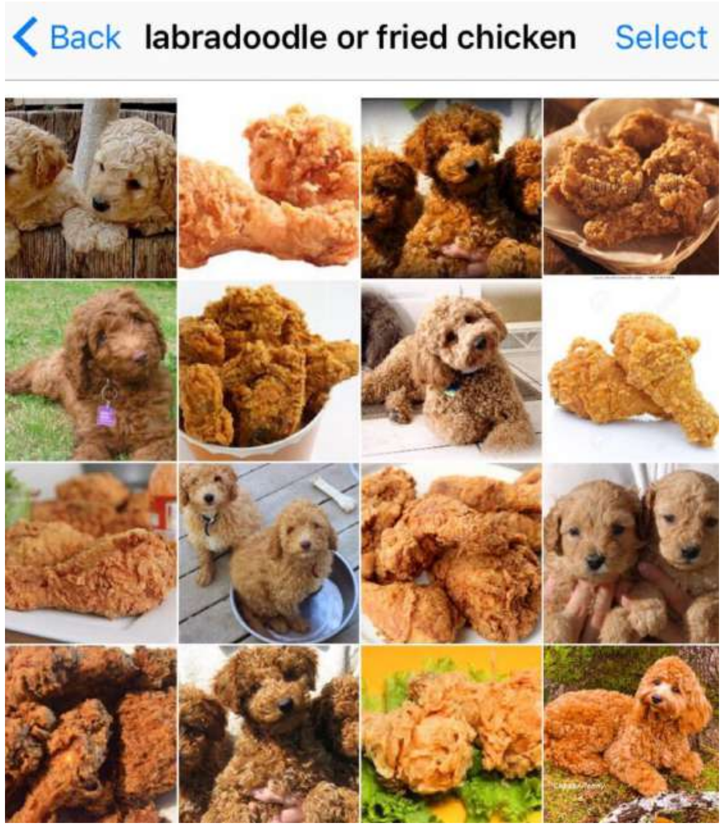
Chihuahua

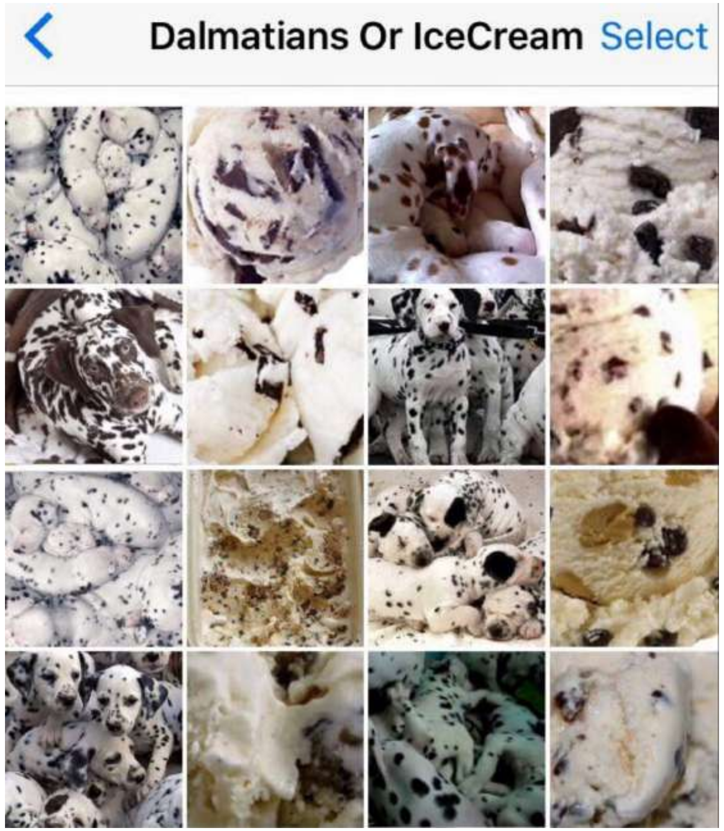
Muffin

… And Lots More!

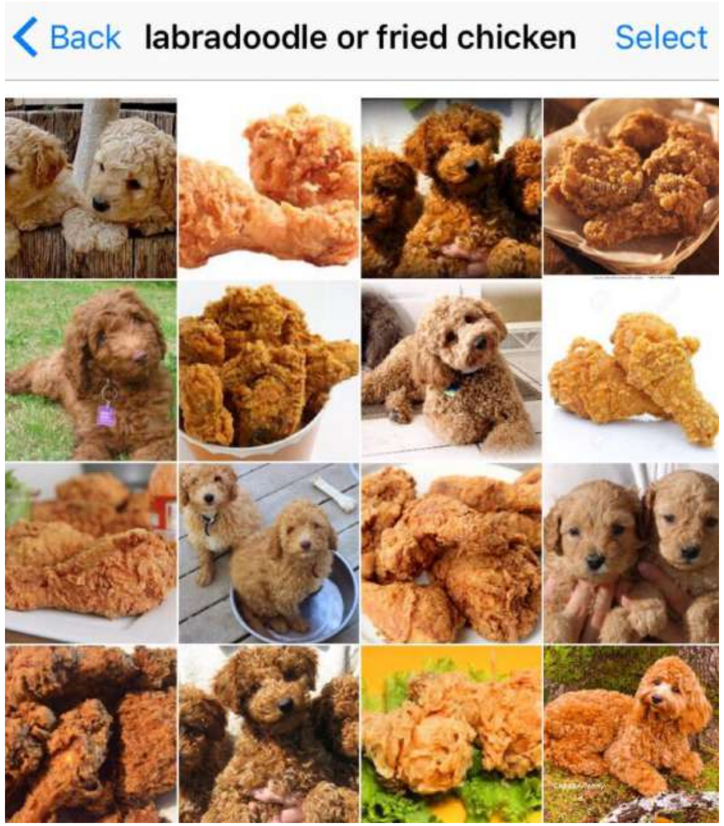


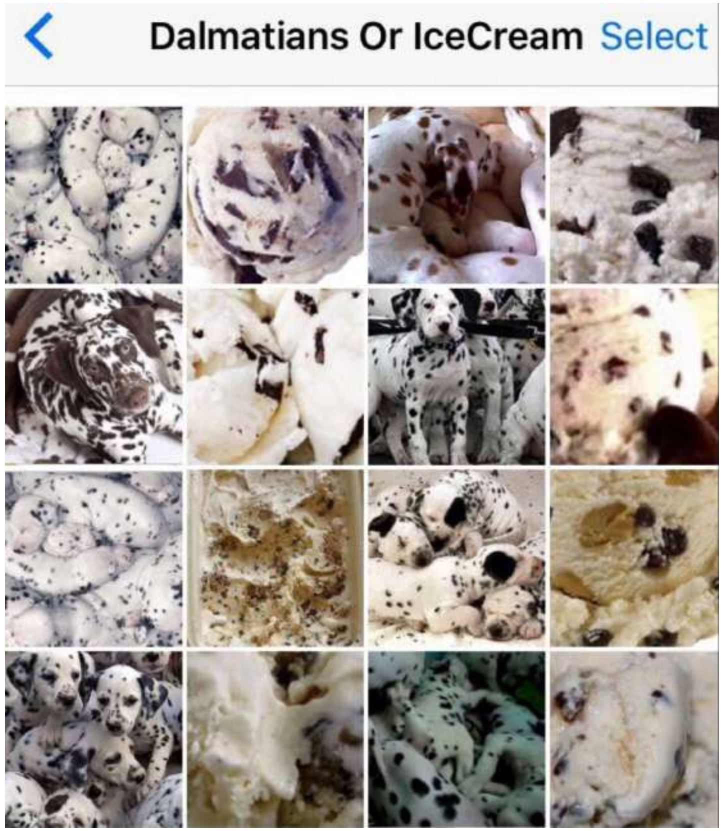
… And Lots More!





… And Lots More!





source: https://www.npr.org/sections/thesalt/2016/03/11/470084215/canine-or-cuisine-this-photo-meme-is-fetching?t=1648392960347

Programming vs. "Training" a Computer

- There exist some problems like the chihuahua vs. muffin above which are too hard to be solved directly

Programming vs. "Training" a Computer

- There exist some problems like the chihuahua vs. muffin above which are too hard to be solved directly
- Hard to design an algorithm which is general enough to capture all the nuances of the problem and gives the correct output for any input

Programming vs. "Training" a Computer

- There exist some problems like the chihuahua vs. muffin above which are too hard to be solved directly
- Hard to design an algorithm which is general enough to capture all the nuances of the problem and gives the correct output for any input

↓

Programming vs. "Training" a Computer

"Training" a Computer

Problem

Labeled Data

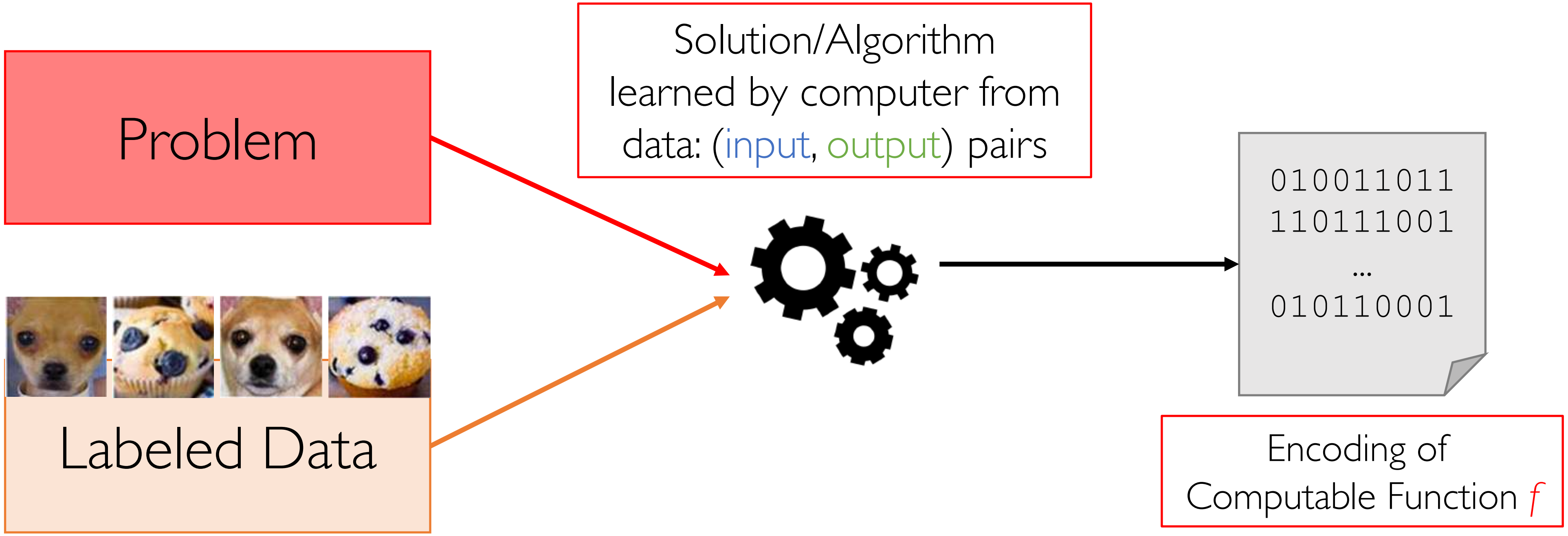
"Training" a Computer

Problem

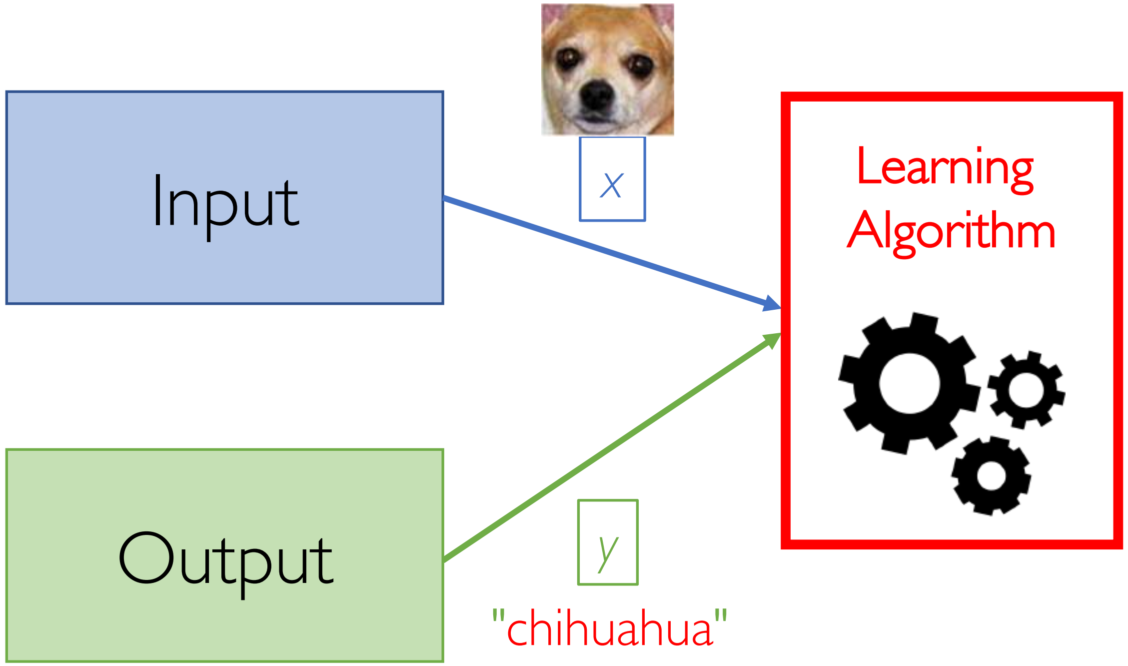
Labeled Data

Solution/Algorithm learned by computer from data: (input, output) pairs

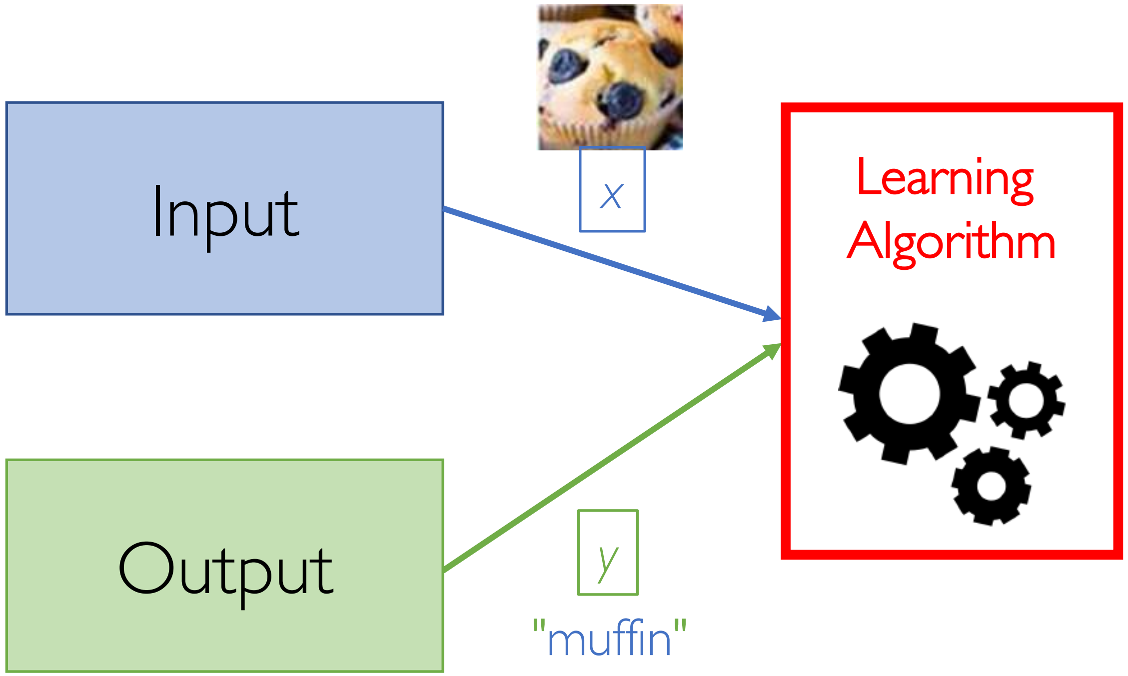
"Training" a Computer



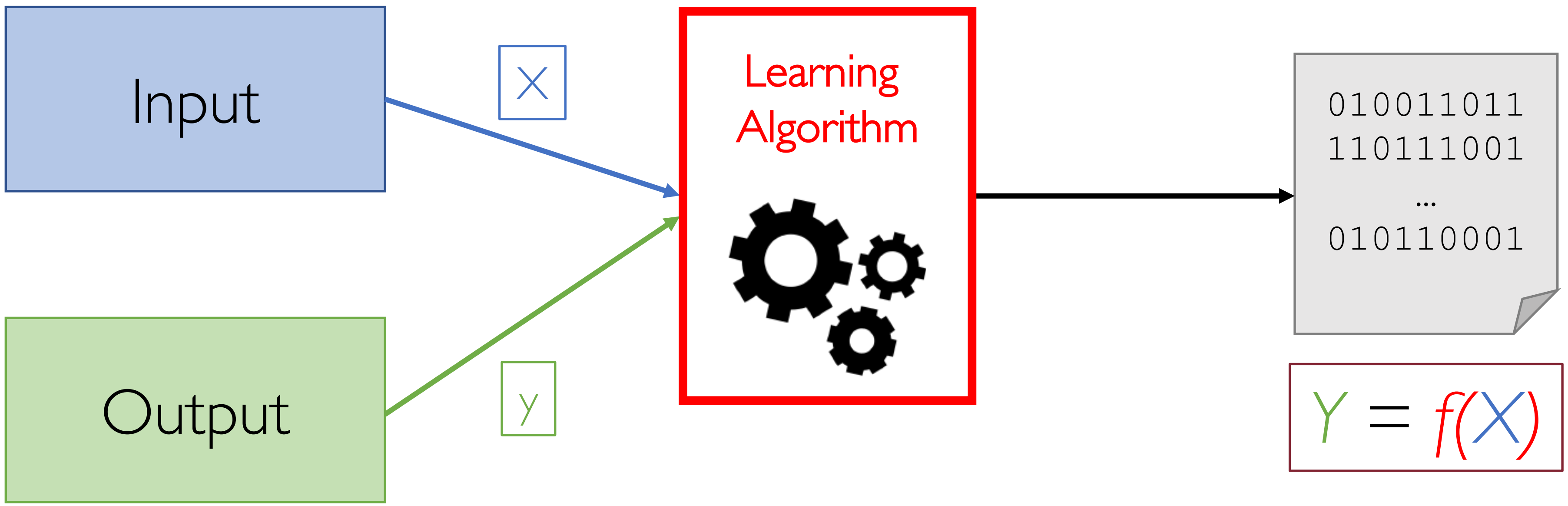
"Training" a Computer



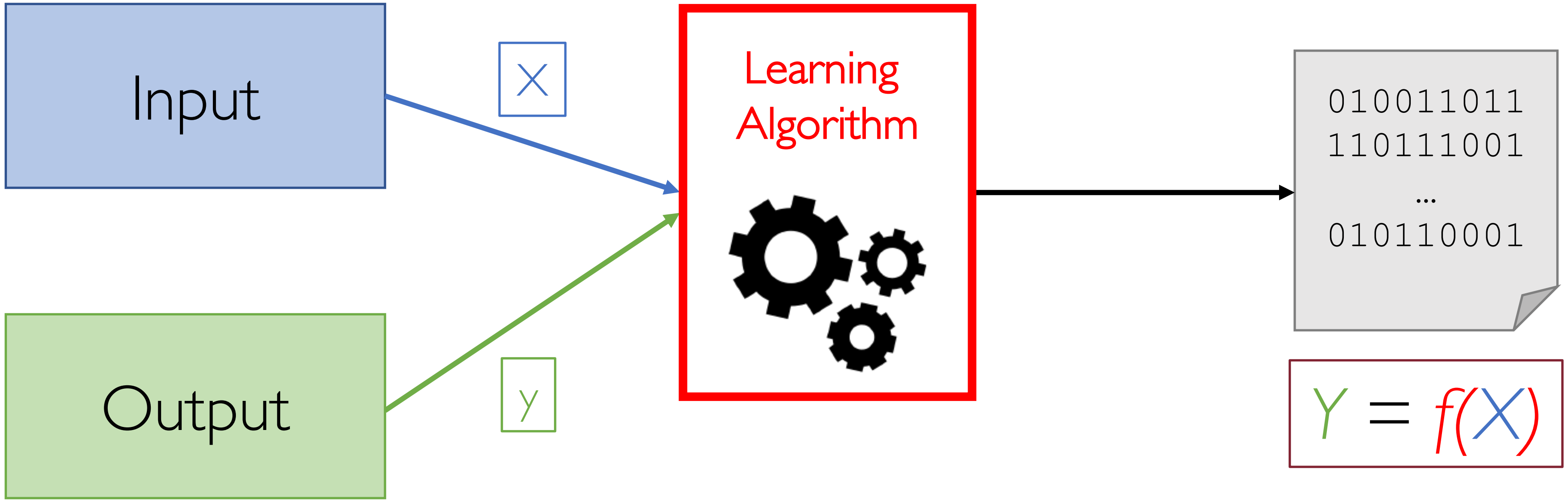
"Training" a Computer



"Training" a Computer



"Training" a Computer



Eventually, the function f is **learned** by the learning algorithm from a (large) set of **labeled data**

Machine Learning

- A broad discipline concerned with how to teach machines to learn (i.e., extract knowledge) from data

Machine Learning

- A broad discipline concerned with how to teach machines to learn (i.e., extract knowledge) from data
- 2 main definitions of it:

Machine Learning

- A broad discipline concerned with how to teach machines to learn (i.e., extract knowledge) from data
- 2 main definitions of it:

> *"The field of study that gives computers the ability to learn without being explicitly programmed"*
>
> Arthur Samuel

Machine Learning

- A broad discipline concerned with how to teach machines to learn (i.e., extract knowledge) from data
- 2 main definitions of it:

"The field of study that gives computers the ability to learn without being explicitly programmed"

Arthur Samuel

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E"

Tom Mitchell

Machine Learning: Taxonomy

Machine Learning

Machine Learning: Taxonomy

Machine Learning

Unsupervised Learning

Extract patterns from input data without any information on the output (target) variable

Machine Learning: Taxonomy

Machine Learning

Unsupervised Learning

Extract patterns from input data without any information on the output (target) variable

Supervised Learning

Approximate a function $f: X \rightarrow Y$ from a set of observed labeled examples $\{(X, y)\}$

Machine Learning: Taxonomy

Machine Learning

Unsupervised Learning

Extract patterns from input data without any information on the output (target) variable

Supervised Learning

Approximate a function $f: X \rightarrow Y$ from a set of observed labeled examples $\{(X, y)\}$

Reinforcement Learning

Use a Reward-Feedback loop to continuously learn and update the hidden behavior or pattern

Machine Learning: Taxonomy

Machine Learning

Unsupervised Learning

Extract patterns from input data without any information on the output (target) variable

Supervised Learning

Approximate a function $f: X \rightarrow Y$ from a set of observed labeled examples $\{(X, y)\}$

Reinforcement Learning

Use a Reward-Feedback loop to continuously learn and update the hidden behavior or pattern

Supervised Learning: What Do We Predict?

Supervised Learning

Supervised Learning: What Do We Predict?

Supervised Learning

Regression

The target y we want to predict is a continuous real value

e.g., y = price of a house

Supervised Learning: What Do We Predict?

Supervised Learning

Regression

The target y we want to predict is a **continuous real value**

e.g., y = price of a house

Classification

The target y we want to predict is a **discrete value**

e.g., y = spam/non-spam

The Supervised Learning Pipeline

The Stages of Supervised Learning

0. Be sure your problem needs <u>actually</u> to be tackled using Machine Learning techniques

(i.e., there is no point in adopting any fancy ML solution if it can be solved "directly"!)

The Stages of Supervised Learning

0. Be sure your problem needs <u>actually</u> to be tackled using Machine Learning techniques
(i.e., there is no point in adopting any fancy ML solution if it can be solved "directly"!)

1. **Data collection:** get data from your domain of interest

The Stages of Supervised Learning

0. Be sure your problem needs <u>actually</u> to be tackled using Machine Learning techniques
(i.e., there is no point in adopting any fancy ML solution if it can be solved "directly"!)

1. **Data collection:** get data from your domain of interest

2. **Feature engineering:** represent data in a "machine-friendly" format

The Stages of Supervised Learning

0. Be sure your problem needs <u>actually</u> to be tackled using Machine Learning techniques
(i.e., there is no point in adopting any fancy ML solution if it can be solved "directly"!)

1. **Data collection:** get data from your domain of interest

2. **Feature engineering:** represent data in a "machine-friendly" format

3. **Model training:** "build" one (or more) learning models

The Stages of Supervised Learning

0. Be sure your problem needs actually to be tackled using Machine Learning techniques
(i.e., there is no point in adopting any fancy ML solution if it can be solved "directly"!)

1. **Data collection:** get data from your domain of interest

2. **Feature engineering:** represent data in a "machine-friendly" format

3. **Model training:** "build" one (or more) learning models

4. **Model selection/evaluation:** pick the best-performing model according to some quality metrics

Data Collection

- Any ML technique needs data to operate on!

Data Collection

- Any ML technique needs data to operate on!
- Supervised Learning requires labeled data which may be even harder to get
  - e.g., emails + spam/non-spam tags

Data Collection

- Any ML technique needs data to operate on!
- Supervised Learning requires labeled data which may be even harder to get
  - e.g., emails + spam/non-spam tags
- Might involve combining multiple and heterogeneous data sources

Feature Engineering

Domain Objects

Feature Engineering

Collected data need to be encoded with a machine-readable format

Domain Objects

Feature Engineering

Collected data need to be encoded with a machine-readable format

Domain Objects

i-th object

0.27	3.8	1.12	…	4.75	…	…	56.9
0.0	3.65	0.04	…	6.81	…	…	1.0

…

6.94	5.4	0.9	...	3.92	...	...	8.2

Each domain object is translated into a *n*-dimensional vector of features

Feature Engineering

Collected data need to be encoded with a machine-readable format

Domain Objects

i-th object

0.27	3.8	1.12	…	4.75	…	…	56.9
0.0	3.65	0.04	…	6.81	…	…	1.0
				…			
6.94	5.4	0.9	...	3.92	...	...	8.2

j-th feature

Each domain object is translated into a *n*-dimensional vector of features

Feature Engineering

- Each feature is a property of an instance of our domain
  - e.g., **number_of_bedrooms** in the case our domain objects are "houses"

Feature Engineering

- Each feature is a property of an instance of our domain
  - e.g., `number_of_bedrooms` in the case our domain objects are "houses"
- Each feature can be either derived locally from an instance
  - e.g., `annual_income` of a person

Feature Engineering

- Each feature is a property of an instance of our domain
  - e.g., `number_of_bedrooms` in the case our domain objects are "houses"
- Each feature can be either derived locally from an instance
  - e.g., `annual_income` of a person
- Or it can be the result of more complex computation involving the whole data collection
  - e.g., `tf-idf` of a word of a document w.r.t. a corpus

Feature Engineering

- Traditionally done manually by human experts

Feature Engineering

- Traditionally done manually by human experts
- Require in-depth knowledge of the specific domain of application
  - e.g., text, images, finance, etc.

Feature Engineering

- Traditionally done manually by human experts
- Require in-depth knowledge of the specific domain of application
  - e.g., text, images, finance, etc.
- Tedious and time-consuming process

Feature Engineering

- Traditionally done manually by human experts
- Require in-depth knowledge of the specific domain of application
  - e.g., text, images, finance, etc.
- Tedious and time-consuming process
- Techniques to **automatically** learn data representation (i.e., features):
  - K-means clustering, PCA, autoencoders (**unsupervised**)
  - Neural Networks (**supervised**)

Feature Engineering: Challenges and Solutions

Collected (raw) data is far from being perfect!

Feature Engineering: Challenges and Solutions

Collected (raw) data is far from being perfect!

Many challenges need to be addressed <u>before</u> taking any further step down to the ML pipeline

Feature Engineering: Challenges and Solutions

Collected (raw) data is far from being perfect!

Many challenges need to be addressed before taking any further step down to the ML pipeline

⇩

Data Preprocessing

Feature Engineering: Challenges and Solutions

Challenge	Description	
Missing values	A feature value may not be available for one or more instances	

Feature Engineering: Challenges and Solutions

Challenge	Description	Solution
Missing values	A feature value may not be available for one or more instances	Replace missing values with the **median** (continuous) or the **mode** (categorical) of the existing values

Feature Engineering: Challenges and Solutions

Challenge	Description	
Sparsity	Most of the instances contain just a small subset of the features	

Feature Engineering: Challenges and Solutions

Challenge	Description	Solution
Sparsity	Most of the instances contain just a small subset of the features	Use "sparse-friendly" data structures (e.g., DOK)

Feature Engineering: Challenges and Solutions

Challenge	Description	
Outliers	One or more instances have out-of-range values for one or more features	

Feature Engineering: Challenges and Solutions

Challenge	Description	Solution
Outliers	One or more instances have out-of-range values for one or more features	Retention vs. Exclusion (*trimming* or *winsorising*)

Feature Engineering: Challenges and Solutions

Challenge	Description	
Mix of continuous and discrete values	Feature set contains both numerical and categorical values	

Feature Engineering: Challenges and Solutions

Challenge	Description	Solution
Mix of continuous and discrete values	Feature set contains both numerical and categorical values	Transform categorical features using *one-hot encoding*

Feature Engineering: Challenges and Solutions

Challenge	Description	
Multiple feature magnitudes	Feature set contains very wide range of values	

Feature Engineering: Challenges and Solutions

Challenge	Description	Solution
Multiple feature magnitudes	Feature set contains very wide range of values	Standardization (min-max, z-scores)

Feature Engineering: Challenges and Solutions

Challenge	Description	
Class imbalance	Instances labeled with the class of interest represents a tiny fraction of the total	

Feature Engineering: Challenges and Solutions

Challenge	Description	Solution
Class imbalance	Instances labeled with the class of interest represents a tiny fraction of the total	Over-/Under-sampling, cost-sensitive learning

Feature Engineering: Challenges and Solutions

Challenge	Description	
Strong multicollinearity	Linear relationship between one or more features	

Feature Engineering: Challenges and Solutions

Challenge	Description	Solution
Strong multicollinearity	Linear relationship between one or more features	Dimensionality reduction (PCA)