SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

10.020 Data Driven World

Supervised Learning

Peng Song, ISTD

Week 6, Lesson 3, 2021

Revision: Working with Data

Python read and manipulate data in numerical tables using **pandas**.

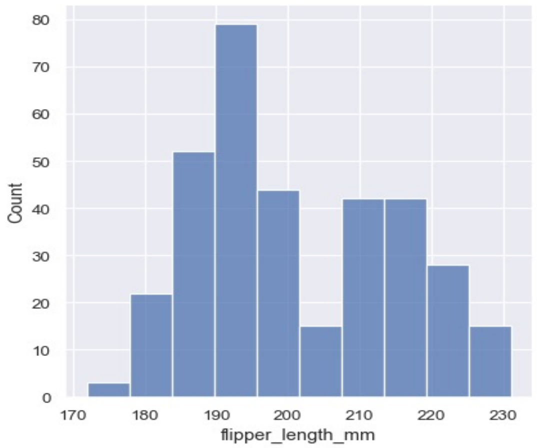
	month	town	flat_type	block	street_name	storey_range	floor_area_sqm	flat_model	lease_commence_date	remaining_lease	resale_price
0	2017-01	ANG MO KIO	2 ROOM	406	ANG MO KIO AVE 10	10 TO 12	44.0	Improved	1979	61 years 04 months	232000.0
1	2017-01	ANG MO KIO	3 ROOM	108	ANG MO KIO AVE 4	01 TO 03	67.0	New Generation	1978	60 years 07 months	250000.0
2	2017-01	ANG MO KIO	3 ROOM	602	ANG MO KIO AVE 5	01 TO 03	67.0	New Generation	1980	62 years 05 months	262000.0
3	2017-01	ANG MO KIO	3 ROOM	465	ANG MO KIO AVE 10	04 TO 06	68.0	New Generation	1980	62 years 01 month	265000.0
4	2017-01	ANG MO KIO	3 ROOM	601	ANG MO KIO AVE 5	01 TO 03	67.0	New Generation	1980	62 years 05 months	265000.0
...	...	...	...	...	...	...	...	...	...	...	...
95853	2021-04	YISHUN	EXECUTIVE	326	YISHUN RING RD	10 TO 12	146.0	Maisonette	1988	66 years 04 months	650000.0
95854	2021-04	YISHUN	EXECUTIVE	360	YISHUN RING RD	04 TO 06	146.0	Maisonette	1988	66 years 04 months	645000.0
95855	2021-04	YISHUN	EXECUTIVE	326	YISHUN RING RD	10 TO 12	146.0	Maisonette	1988	66 years 04 months	585000.0
95856	2021-04	YISHUN	EXECUTIVE	355	YISHUN RING RD	10 TO 12	146.0	Maisonette	1988	66 years 08 months	675000.0
95857	2021-04	YISHUN	EXECUTIVE	277	YISHUN ST 22	04 TO 06	146.0	Maisonette	1985	63 years 05 months	625000.0

95858 rows × 11 columns

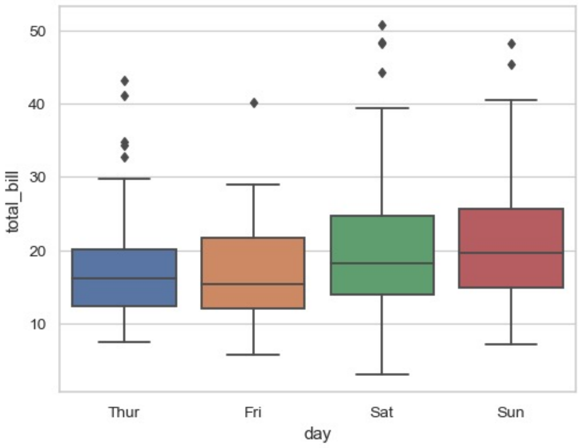
Revision: Data Visualization

Python draw common plots to visualize data using **Matplotlib** and **Seaborn**.

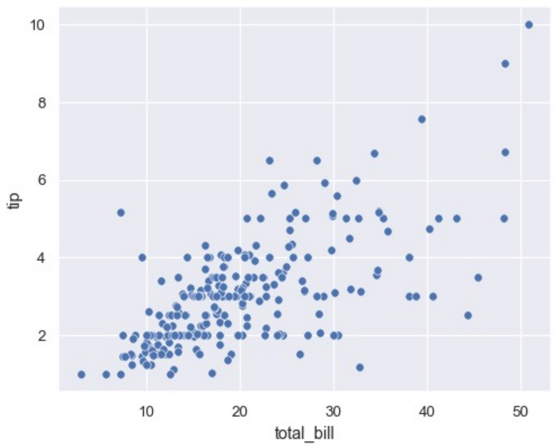
#1 hist plot



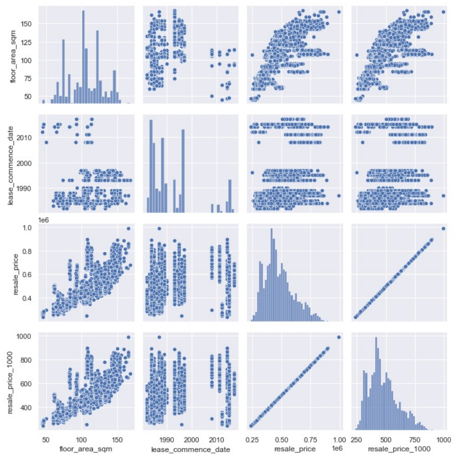
#2 box plot



#3 scatter plot



#4 pair plot



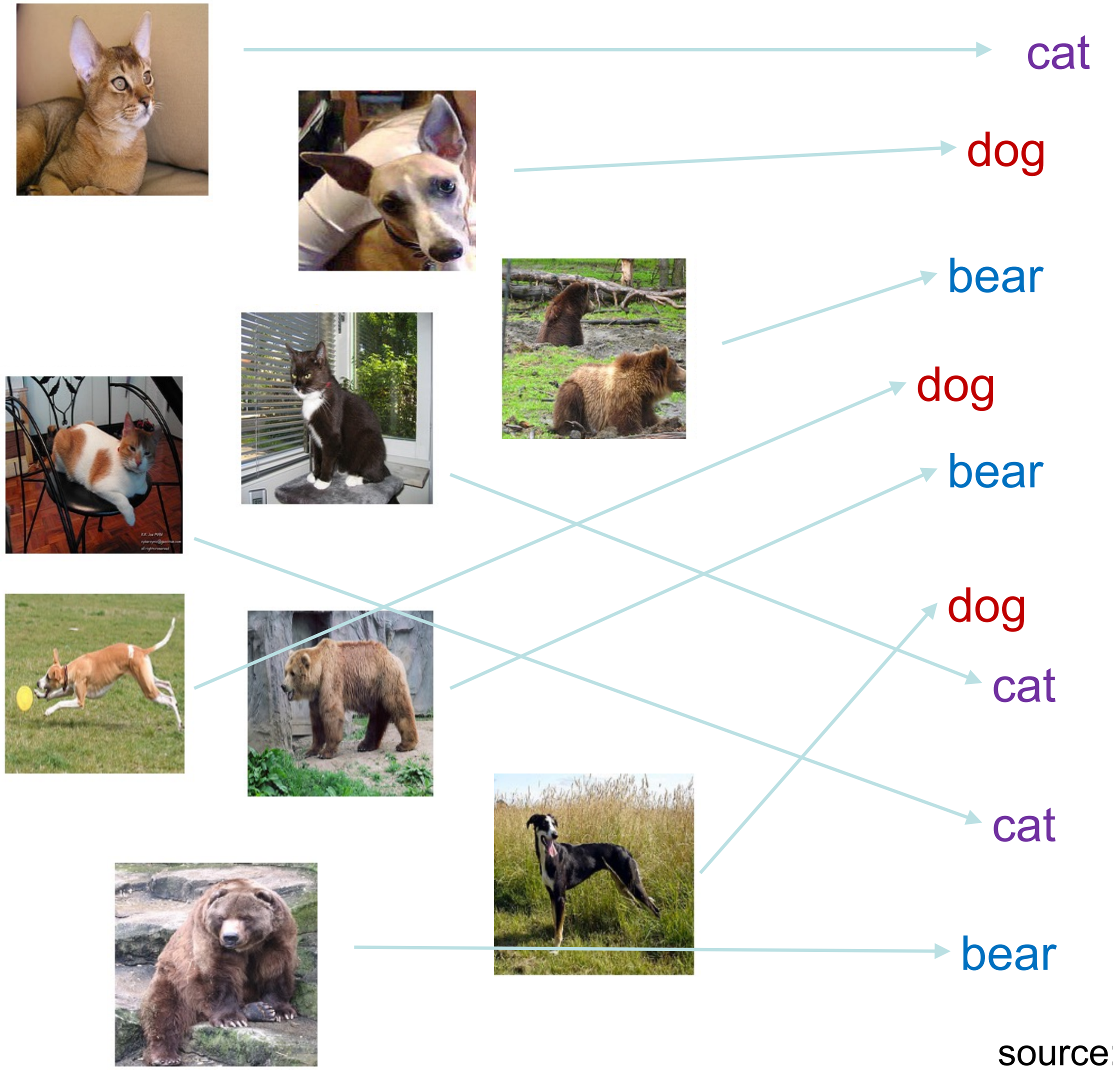
Revision: Types of Machine Learning

Our focus in this course

- **Supervised learning**
  - Given: training data + desired outputs (labels)
- Unsupervised learning
  - Given: training data (without labels)
- Reinforcement learning
  - Rewards from sequence of actions

Supervised Learning vs Unsupervised Learning

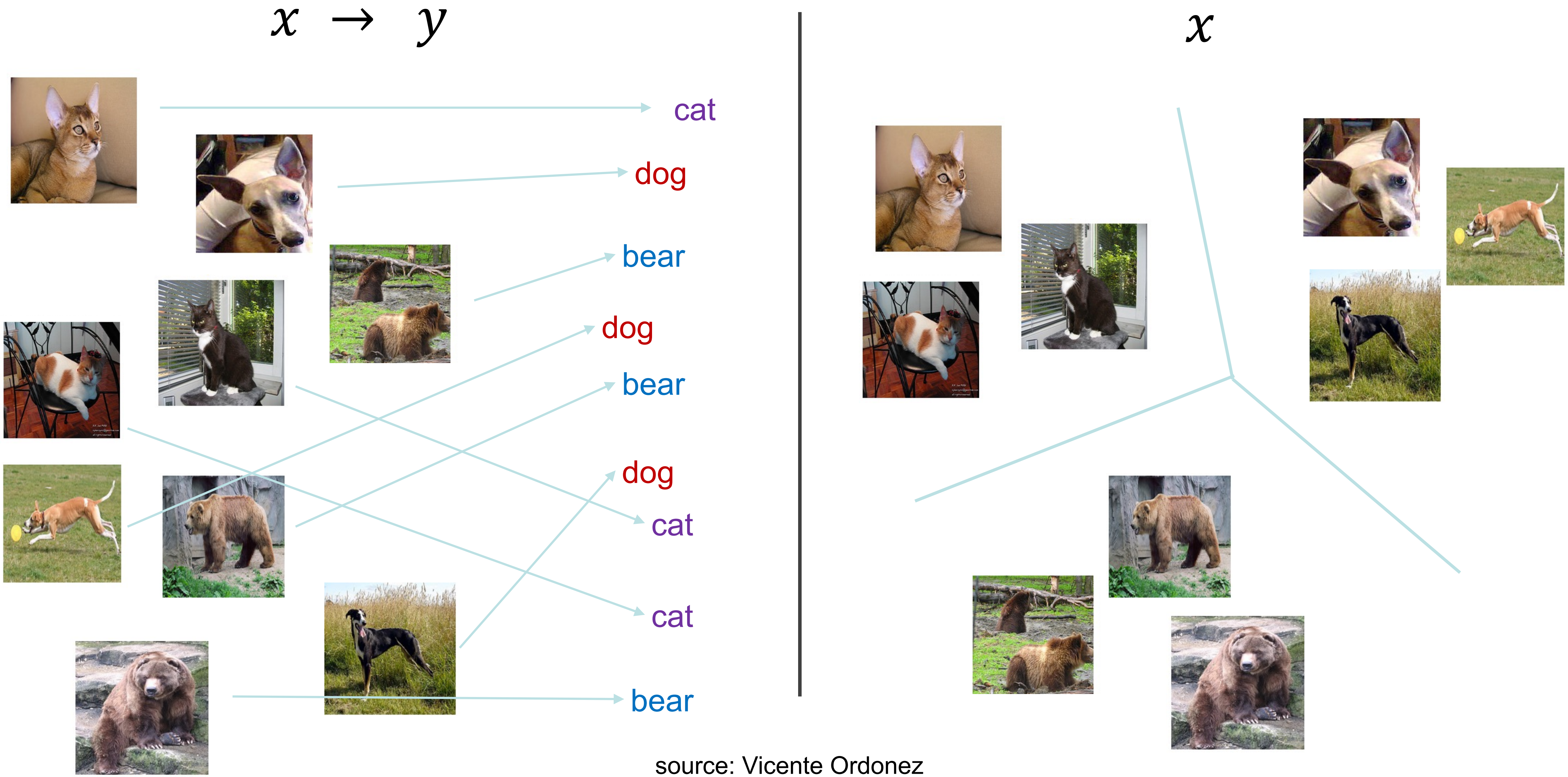
$x \rightarrow y$



x

source: Vicente Ordonez

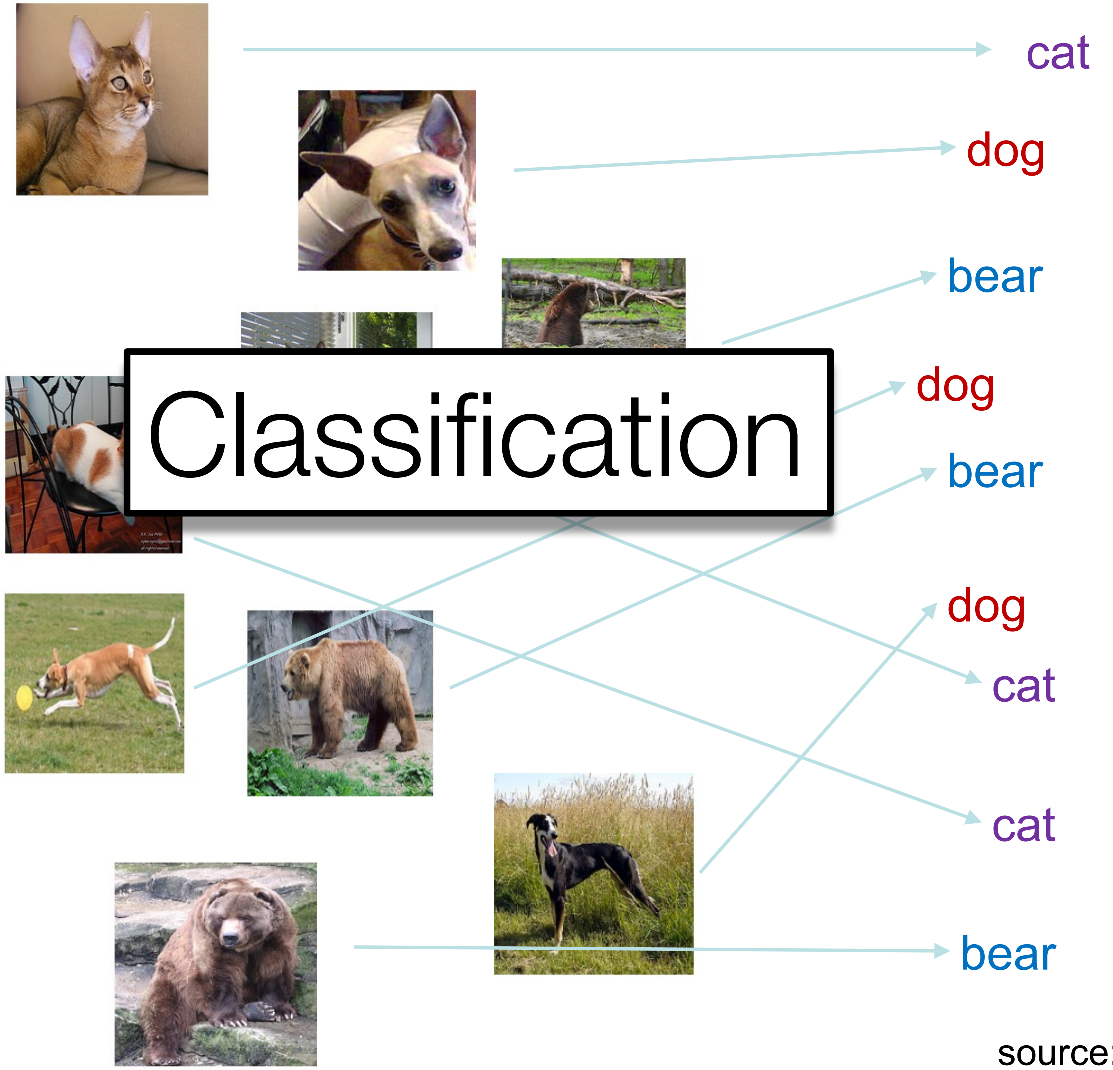
Supervised Learning vs Unsupervised Learning



source: Vicente Ordonez

Supervised Learning vs Unsupervised Learning

$x \to y$

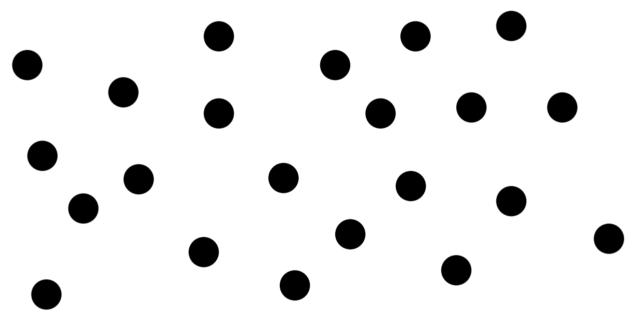


x

source: Vicente Ordonez

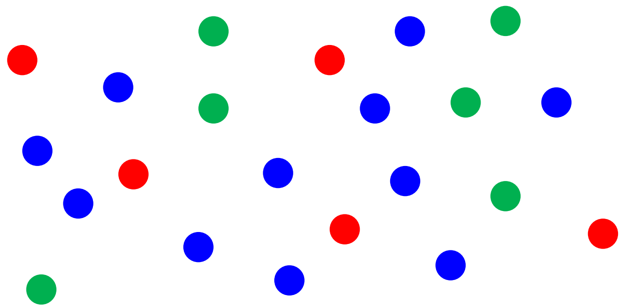
Supervised Learning

- **Classification**
  - Given $(x_1, y_1),\ (x_2, y_2), \dots, (x_n, y_n)$
  - Learn a function $f(x)$ to predict y given x
  - y is **categorical**



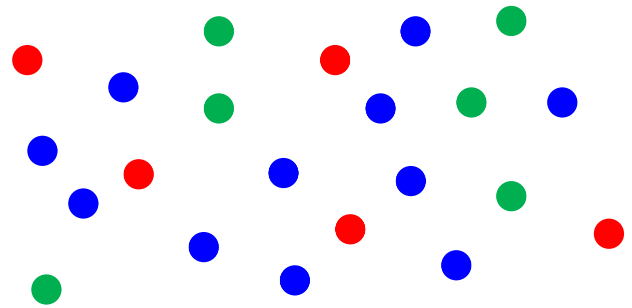
Supervised Learning

- **Classification**
  - Given $(x_1, y_1),\ (x_2, y_2), \dots, (x_n, y_n)$
  - Learn a function $f(x)$ to predict y given x
  - y is **categorical**

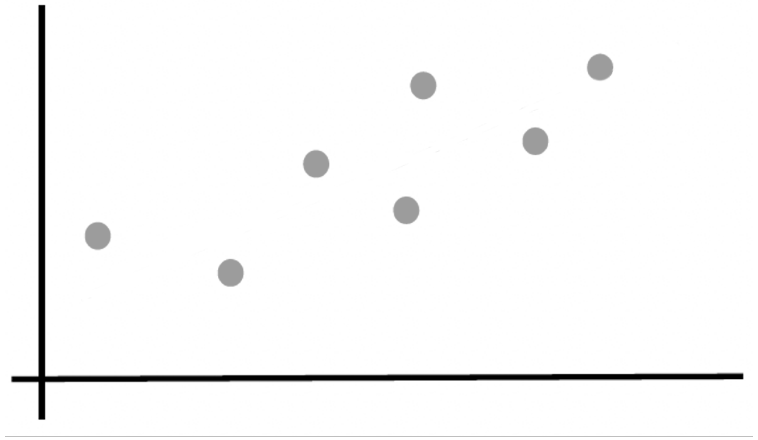


Supervised Learning

- **Classification**
  - Given $(x_1, y_1),\ (x_2, y_2), \ldots, (x_n, y_n)$
  - Learn a function $f(x)$ to predict y given x
  - y is **categorical**

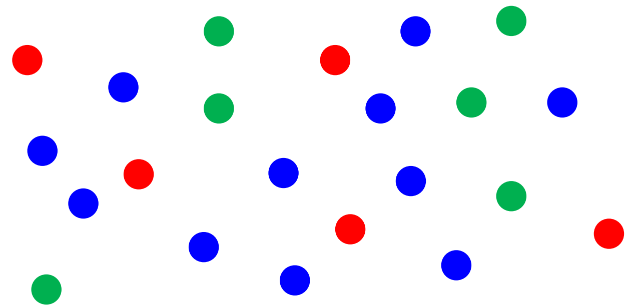


- **Regression**
  - Given $(x_1, y_1),\ (x_2, y_2), \ldots, (x_n, y_n)$
  - Learn a function $f(x)$ to predict y given x
  - y is **numeric**

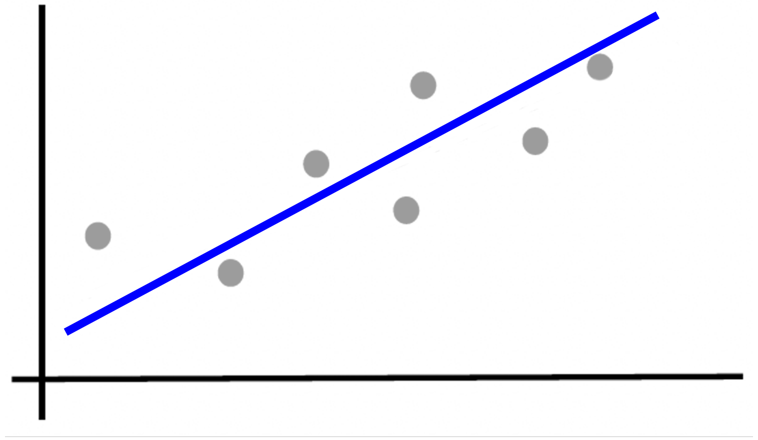


Supervised Learning

- **Classification**
  - Given $(x_1, y_1),\ (x_2, y_2), \dots, (x_n, y_n)$
  - Learn a function $f(x)$ to predict y given x
  - y is **categorical**



- **Regression**
  - Given $(x_1, y_1),\ (x_2, y_2), \dots, (x_n, y_n)$
  - Learn a function $f(x)$ to predict y given x
  - y is **numeric**



Supervised Learning

- **Classification**
  - Given $(x_1, y_1),\ (x_2, y_2), \ldots, (x_n, y_n)$
  - Learn a function $f(x)$ to predict y given x
  - y is **categorical**

Week 10
Logistic Regression

- **Regression**
  - Given $(x_1, y_1),\ (x_2, y_2), \ldots, (x_n, y_n)$
  - Learn a function $f(x)$ to predict y given x
  - y is **numeric**

Week 9
Linear Regression

Classification #1: Animal Recognition

- Represent input image as a vector $\mathbf{x} \in \mathbb{R}^{w \times h \times 3}$
- Learn a classifier $f(\mathbf{x})$ such that,

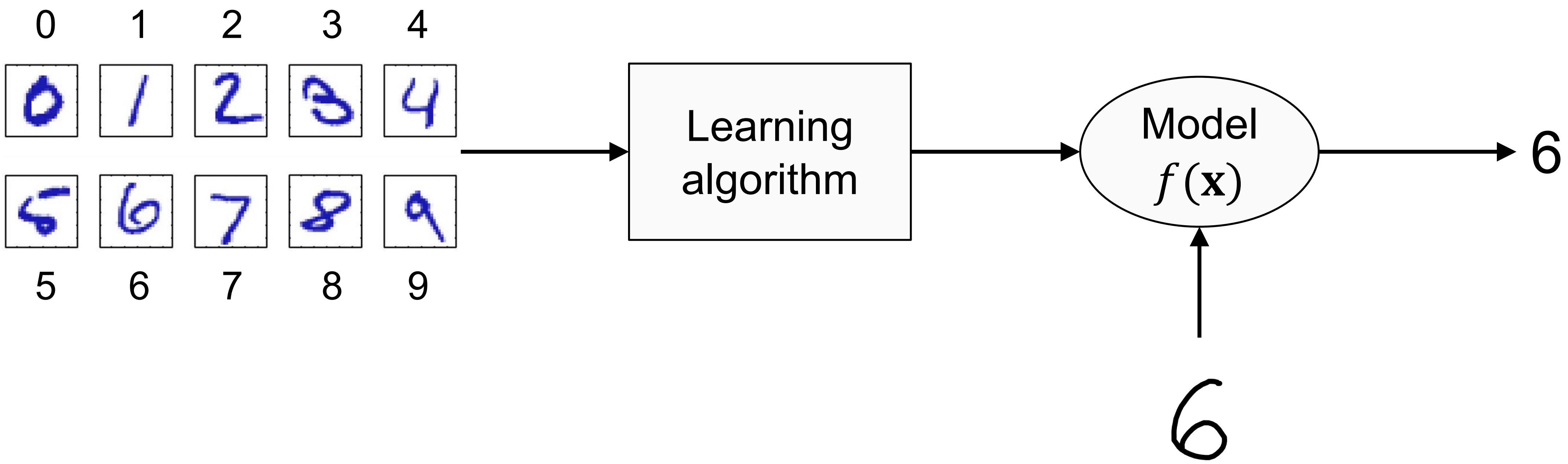
$$f: \mathbf{x} \rightarrow \{\text{zebra}, \text{tiger}, \text{rhino}, \text{panda}, \text{lion}, \text{hippo}, \text{penguin}, \text{giraffe}, \text{snake}, \text{elephant}\}$$

$\rightarrow$ Learning algorithm $\rightarrow$ Model $f(\mathbf{x})$ $\rightarrow$ lion

Classification #2: Hand-written Digit Recognition

- Represent input image as a vector $\mathbf{x} \in \mathbb{R}^{w \times h}$
- Learn a classifier $f(\mathbf{x})$ such that,

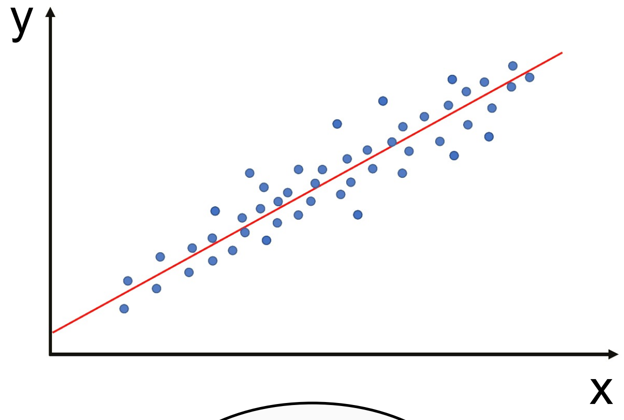
$$f\colon \mathbf{x} \;\rightarrow\; \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

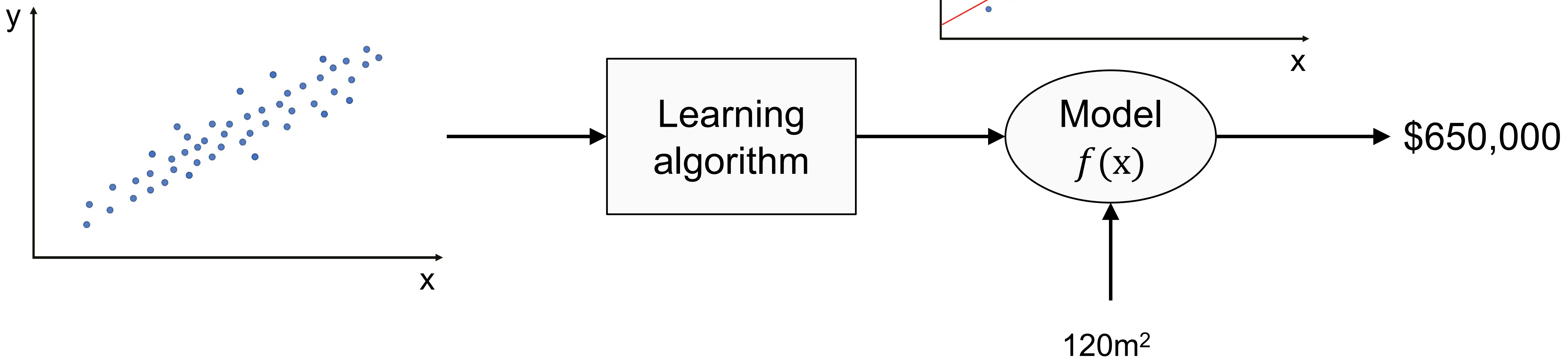


Regression #1: House Price Prediction

- Learn a function $y = f(x)$, where
  - x is house size
  - y is house price
  - f is a linear function

Hypothesis

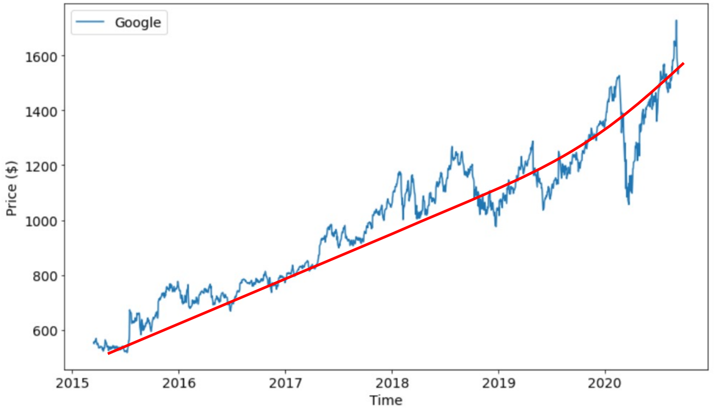


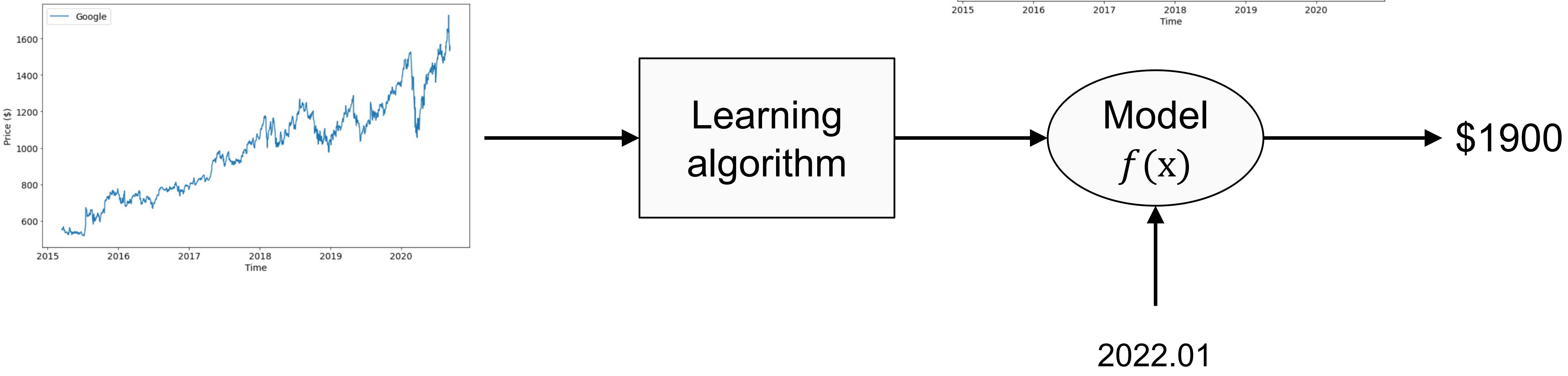


Regression #2: Stock Price Prediction

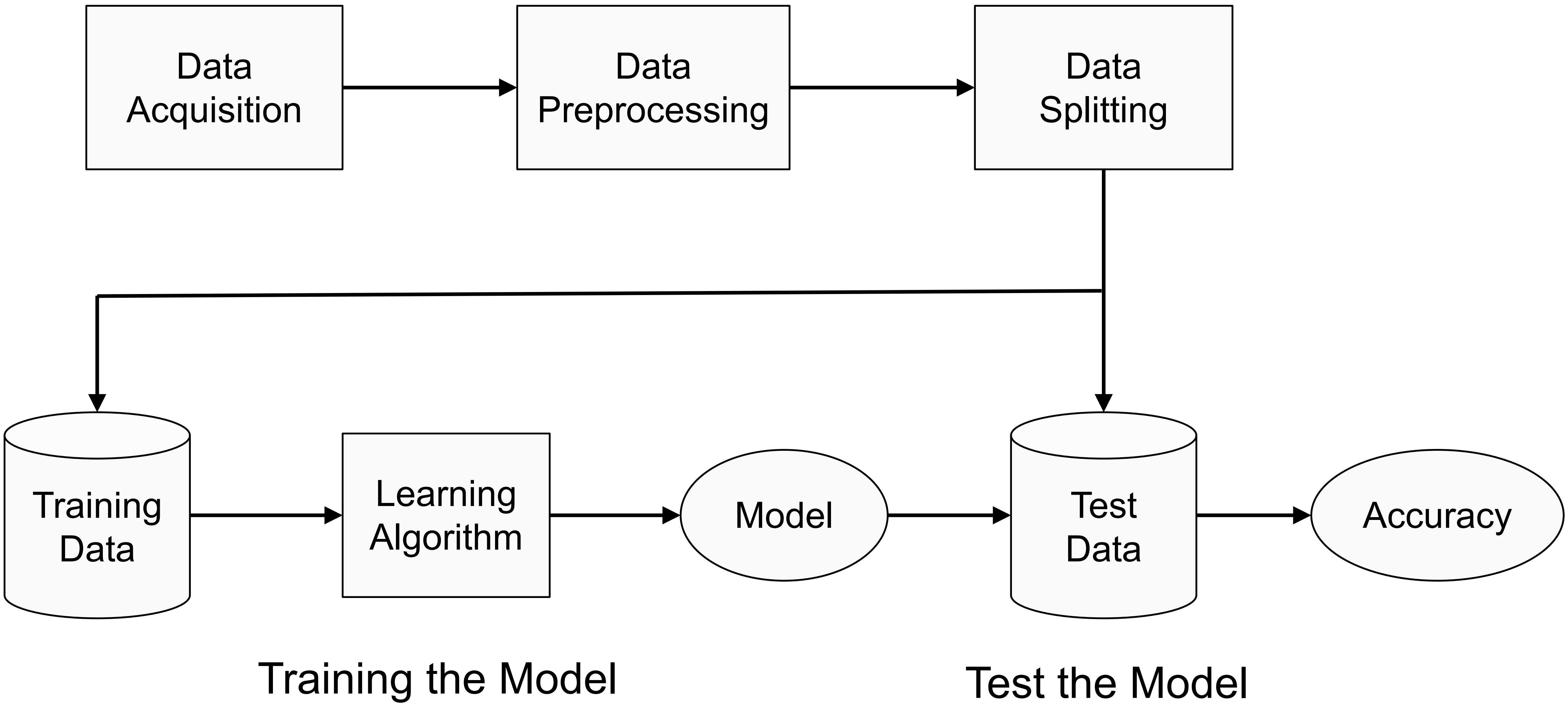
- Learn a function $y = f(x)$, where
  - x is a date
  - y is stock price
  - f is a polynomial function

Hypothesis

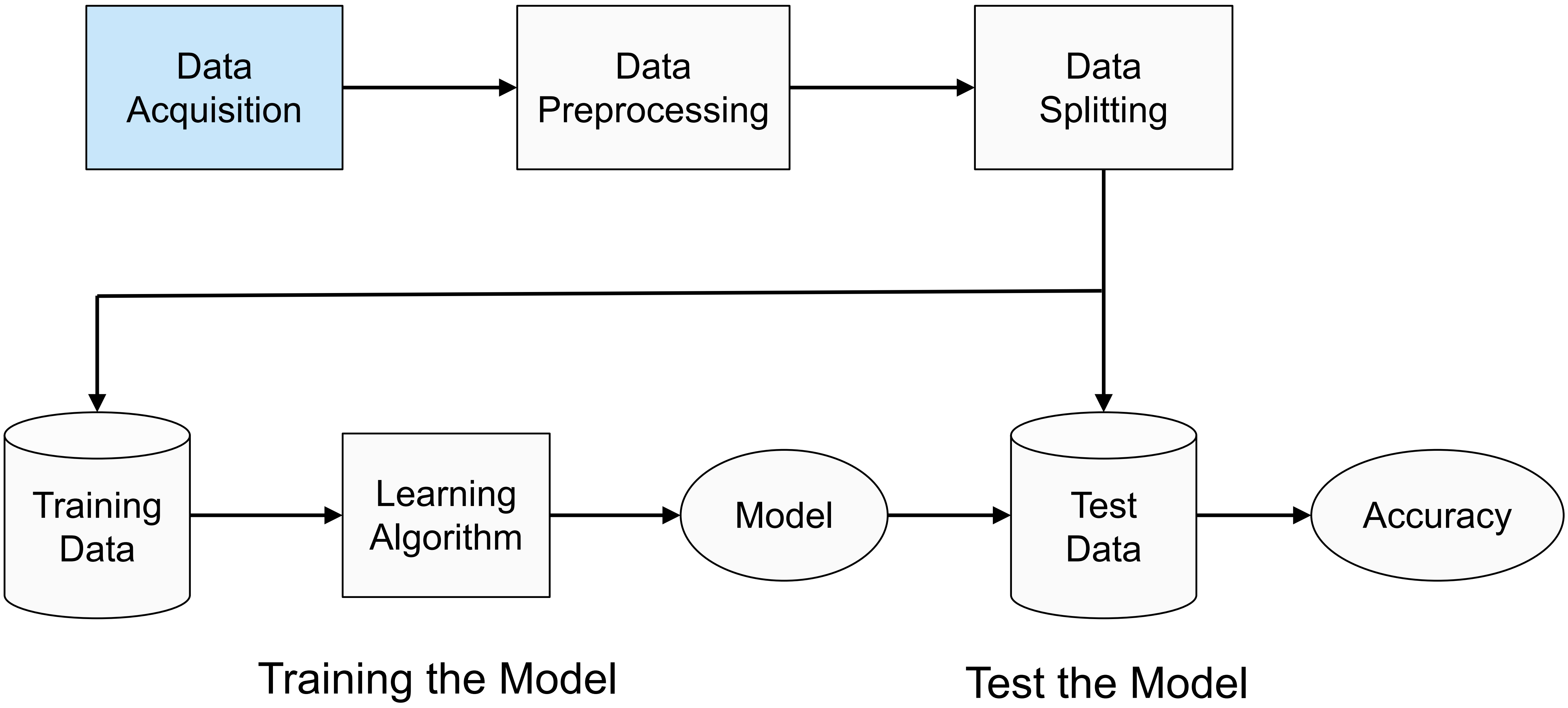




Supervised Learning Process



Supervised Learning Process



Data Acquisition

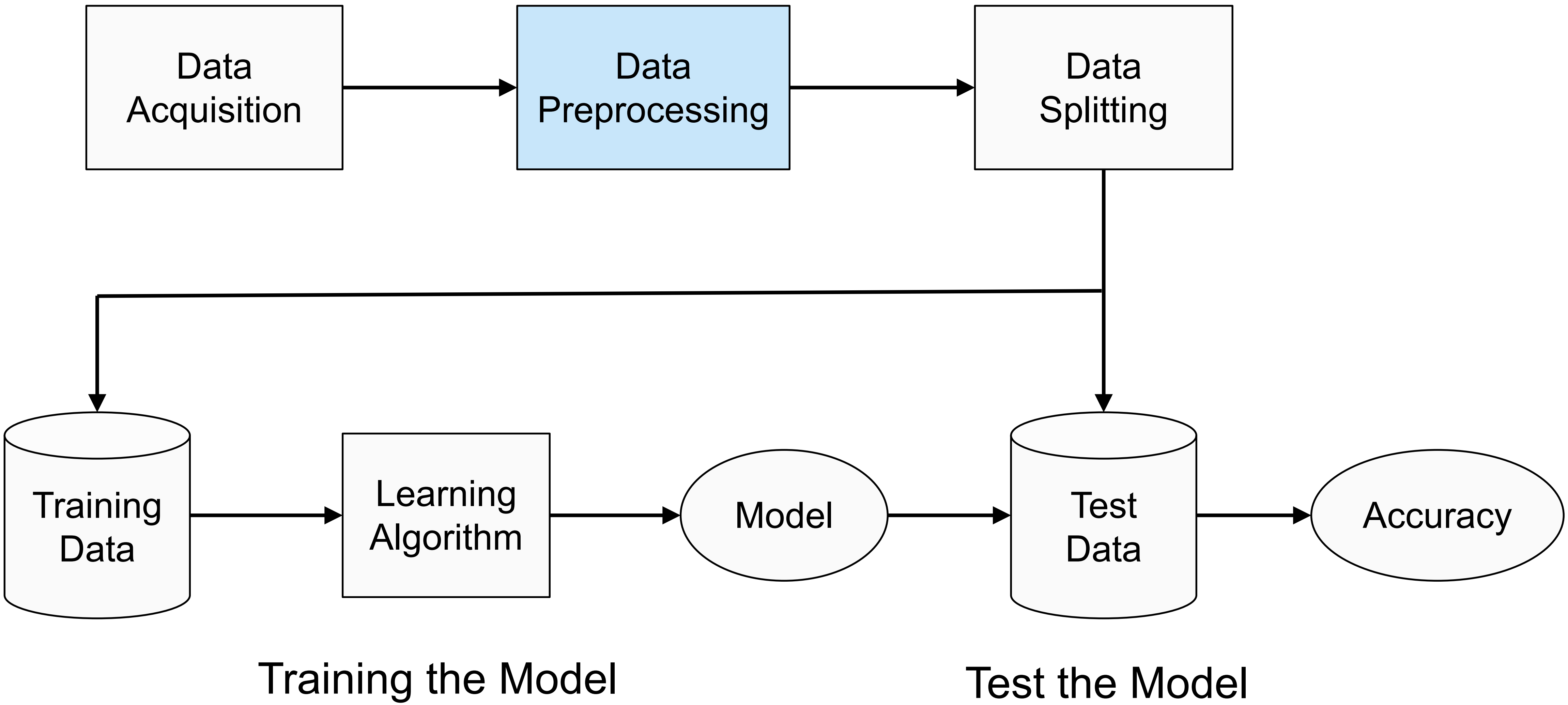
- Data acquisition is the process to acquire datasets that can be used to train the machine learning models.

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	5.33	36.2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273.0	21.0	391.99	9.67	22.4
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273.0	21.0	396.90	9.08	20.6
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273.0	21.0	396.90	5.64	23.9
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273.0	21.0	393.45	6.48	22.0
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273.0	21.0	396.90	7.88	11.9

Data Acquisition Approaches

1. Data Discovery
   – Search for datasets available on the web
2. Data Augmentation
   – Enriching existing data by adding more external data
3. Data Generation
   – Generate the datasets manually or automatically

Supervised Learning Process



Data Extraction

- Extract data for machine learning

INDUS: proportion of non-retail business acres per town
RM: average number of rooms per dwelling
DIS: weighted distances to five Boston employment centers
MEDV: median value of owner-occupied homes in $1000s

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	5.33	36.2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273.0	21.0	391.99	9.67	22.4
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273.0	21.0	396.90	9.08	20.6
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273.0	21.0	396.90	5.64	23.9
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273.0	21.0	393.45	6.48	22.0
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273.0	21.0	396.90	7.88	11.9

Data Extraction

- Extract data for machine learning

INDUS: proportion of non-retail business acres per town
RM: average number of rooms per dwelling
DIS: weighted distances to five Boston employment centers
MEDV: median value of owner-occupied homes in $1000s

features

	RM	DIS	INDUS
0	6.575	4.0900	2.31
1	6.421	4.9671	7.07
2	7.185	4.9671	7.07
3	6.998	6.0622	2.18
4	7.147	6.0622	2.18
...	...	...	...
501	6.593	2.4786	11.93
502	6.120	2.2875	11.93
503	6.976	2.1675	11.93
504	6.794	2.3889	11.93
505	6.030	2.5050	11.93

	MEDV
0	24.0
1	21.6
2	34.7
3	33.4
4	36.2
...	...
501	22.4
502	20.6
503	23.9
504	22.0
505	11.9

target

Data Normalization

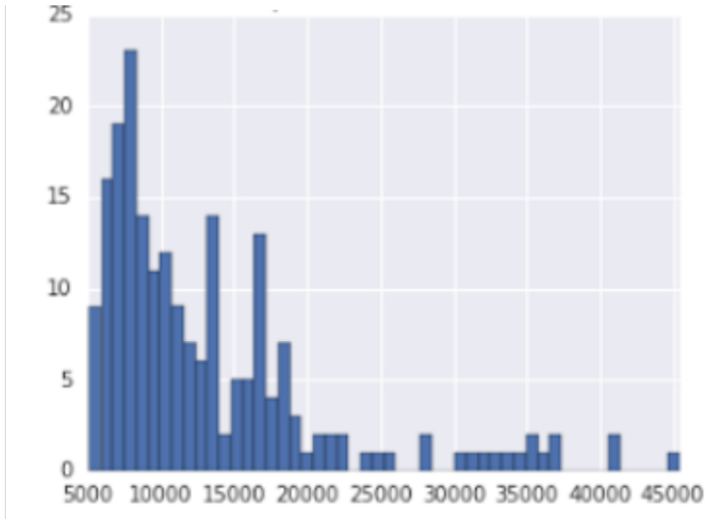
- Minmax normalization
- Z normalization

Minmax Normalization

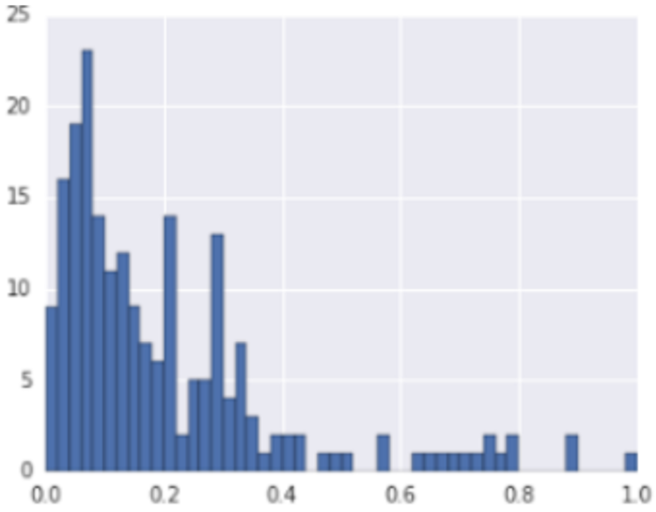
- Linear scale data to range $[0, 1]$

$$normalized = \frac{data \ - min}{max - min}$$

original data



minmax normalization



source

Minmax Normalization

- Linear scale data to range $[0, 1]$

$$normalized = \frac{data\ - min}{max - min}$$

features

	RM	DIS	INDUS
0	6.575	4.0900	2.31
1	6.421	4.9671	7.07
2	7.185	4.9671	7.07
3	6.998	6.0622	2.18
4	7.147	6.0622	2.18
...	...	...	...
501	6.593	2.4786	11.93
502	6.120	2.2875	11.93
503	6.976	2.1675	11.93
504	6.794	2.3889	11.93
505	6.030	2.5050	11.93

target

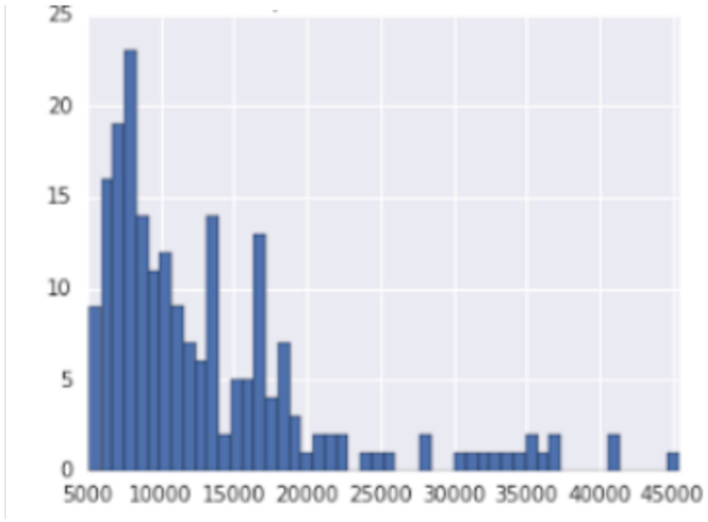
	MEDV
0	24.0
1	21.6
2	34.7
3	33.4
4	36.2
...	...
501	22.4
502	20.6
503	23.9
504	22.0
505	11.9

Z Normalization

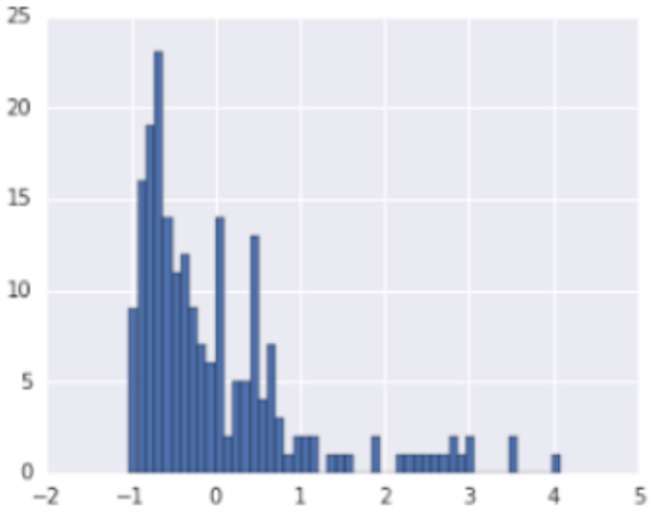
- Linear scale data such that the average is 0 and the standard deviation is 1

$$normalized = \frac{data\ -\ \mu}{\sigma}$$

original data



z normalization



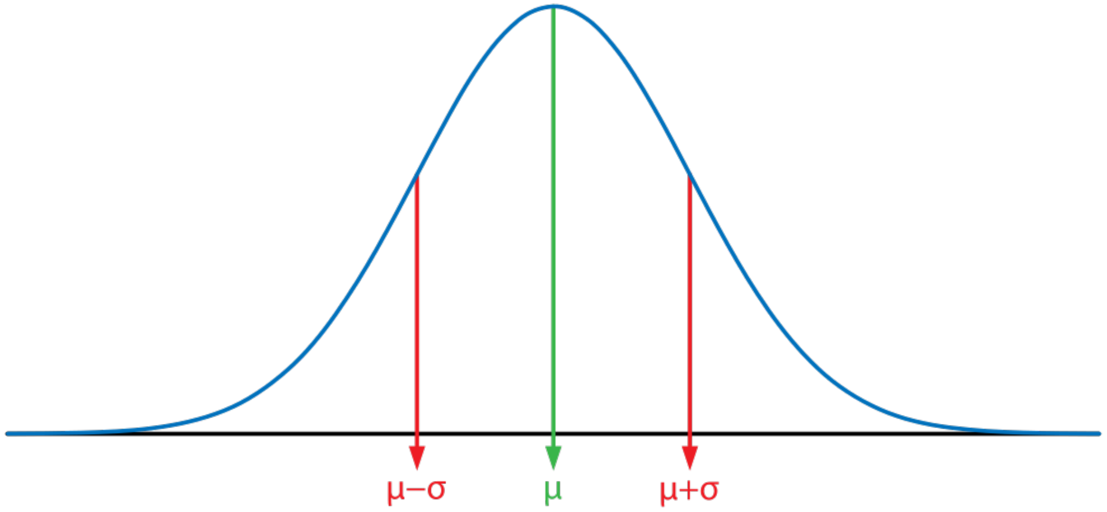
source

Z Normalization

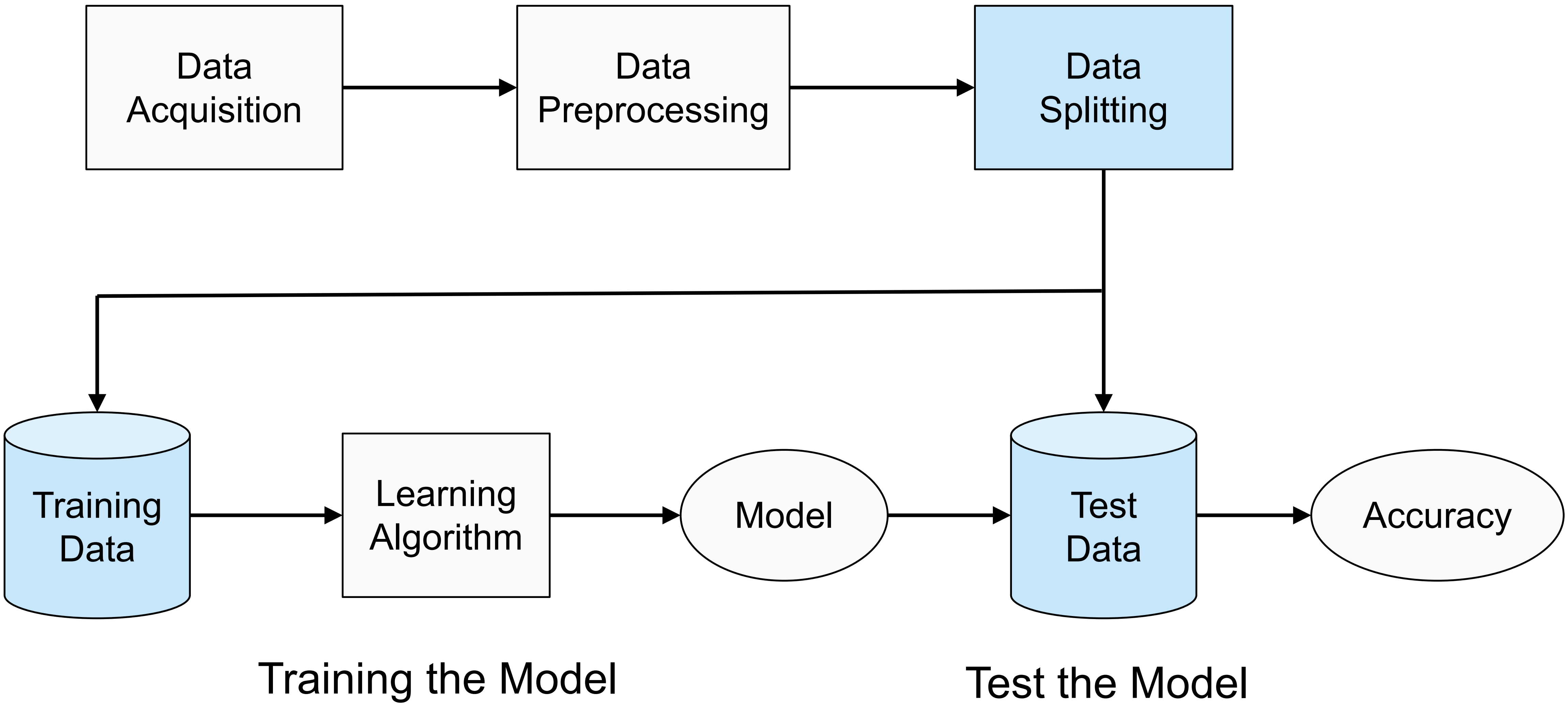
- Linear scale data such that the average is 0 and the standard deviation is 1

$$normalized = \frac{data\ -\ \mu}{\sigma}$$

- Assumption: the data has a Gaussian distribution



Supervised Learning Process



Data Splitting

- Split the data into:
  - **training** dataset
  - **test** dataset

Would this be a good way to do the data splitting?

features

	RM	DIS	INDUS
0	6.575	4.0900	2.31
1	6.421	4.9671	7.07
2	7.185	4.9671	7.07
3	6.998	6.0622	2.18
4	7.147	6.0622	2.18
...	...	...	...
501	6.593	2.4786	11.93
502	6.120	2.2875	11.93
503	6.976	2.1675	11.93
504	6.794	2.3889	11.93
505	6.030	2.5050	11.93

target

	MEDV
0	24.0
1	21.6
2	34.7
3	33.4
4	36.2
...	...
501	22.4
502	20.6
503	23.9
504	22.0
505	11.9

Data Splitting

- Split the data into:
  - **training** dataset
  - **test** dataset

- The split must be done **randomly** to avoid systematic bias in the split of the dataset.

features

	RM	DIS	INDUS
0	6.575	4.0900	2.31
1	6.421	4.9671	7.07
2	7.185	4.9671	7.07
3	6.998	6.0622	2.18
4	7.147	6.0622	2.18
...	...	...	...
501	6.593	2.4786	11.93
502	6.120	2.2875	11.93
503	6.976	2.1675	11.93
504	6.794	2.3889	11.93
505	6.030	2.5050	11.93

target

	MEDV	
0	24.0	train
1	21.6	test
2	34.7	test
3	33.4	train
4	36.2	train
...	...	
501	22.4	test
502	20.6	train
503	23.9	test
504	22.0	train
505	11.9	train

Fundamental Assumption

- **Assumption**: The distribution of training examples is identical to the distribution of test examples (including future unseen examples).
  - In practice, this assumption is often violated to certain degree.
  - Strong violations will clearly result in poor prediction accuracy.

- To achieve good accuracy on the test data, training examples must be sufficiently representative of the test data.

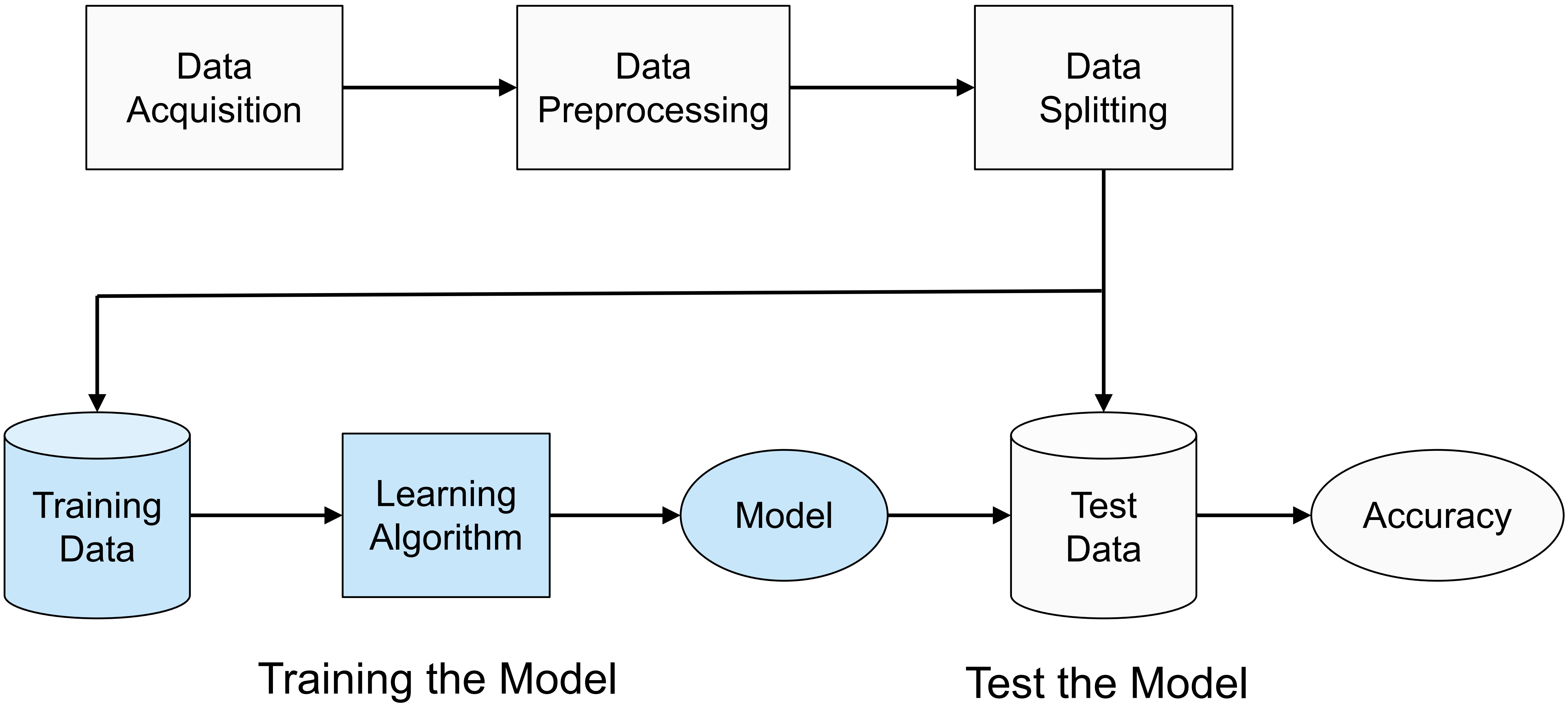
Data Splitting Percentage

- The procedure has one main configuration parameter, which is the size of the train and test sets.
- This is most commonly expressed as a percentage between 0 and 1 for either the train or test datasets, e.g.,
  - Train: 80%, Test: 20%
  - Train: 67%, Test: 33%
  - Train: 50%, Test: 50%

Data Splitting: Tuning the Model

- There are times in machine learning, we need to experiment with different parameters and find the optimum parameters.
- In these cases, the dataset is usually split into three:
  - **training** dataset, which is used to build the model
  - **validation** dataset, which is used to evaluate the model for various parameters and to choose the optimum parameter
  - **test** dataset, which is used to evaluate the model built with the optimum parameter found previously

Supervised Learning Process



Training the Model

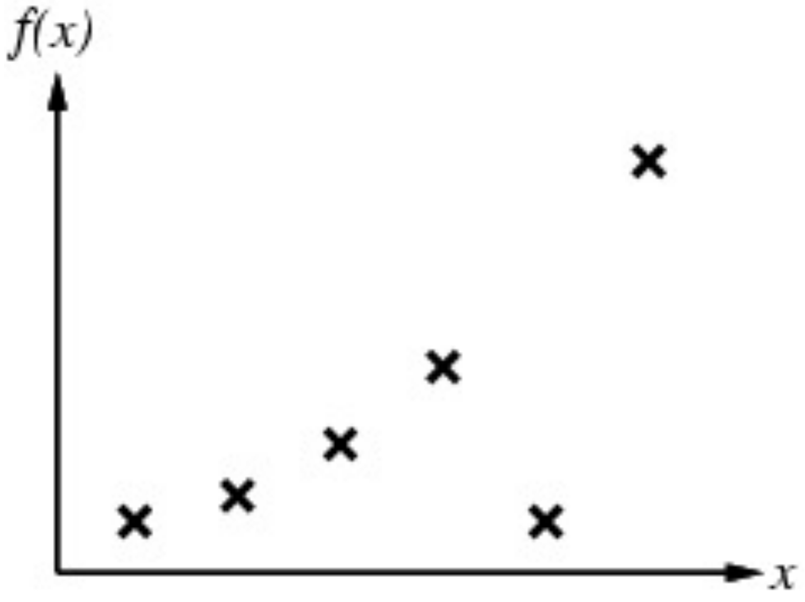
- Simplest form: learn a function from examples
  - *f* is the target function
  - An example is a pair (x, $f(x)$)

- Pure induction task:
  - Given a collection of examples of *f*, return a function *h* that approximates *f*.
  - find a hypothesis *h*, such that $h \approx f$, given a training set of examples

- This is a highly simplified model of real learning:
  - Ignores prior knowledge
  - Assumes examples are given

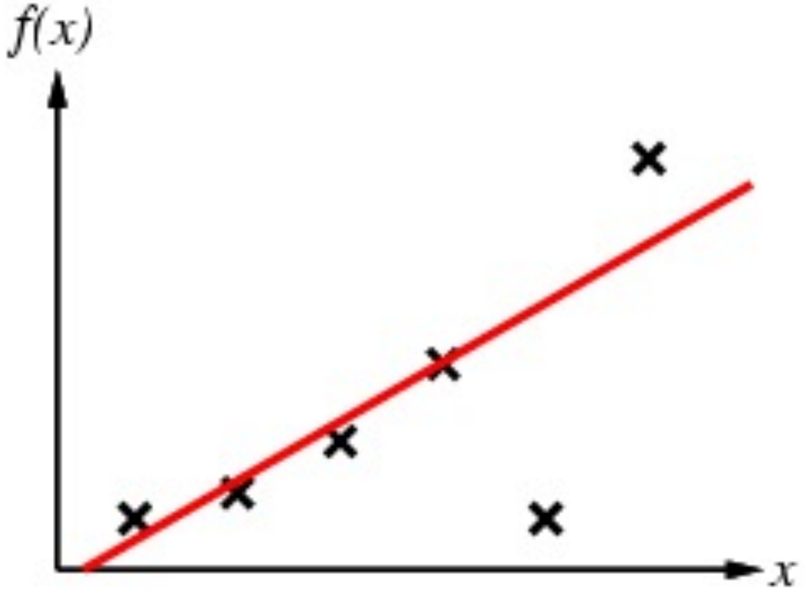
Training the Model

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



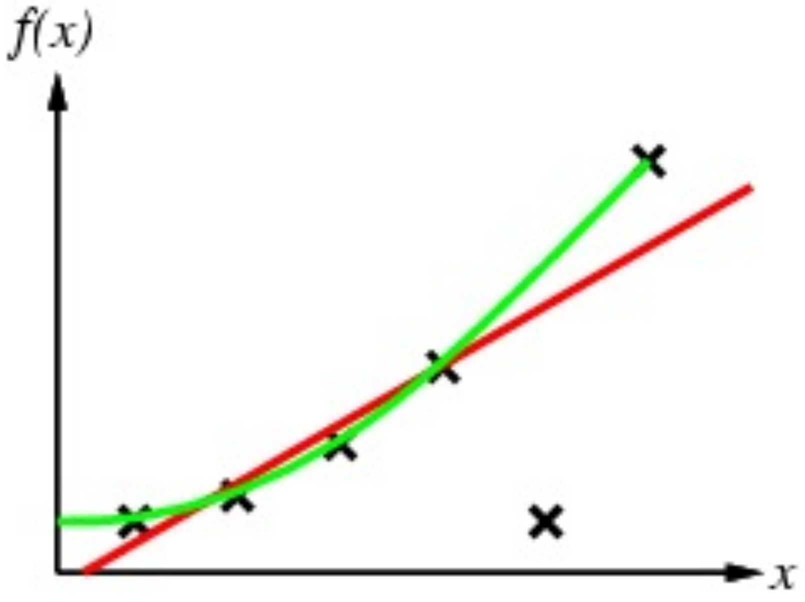
Training the Model

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



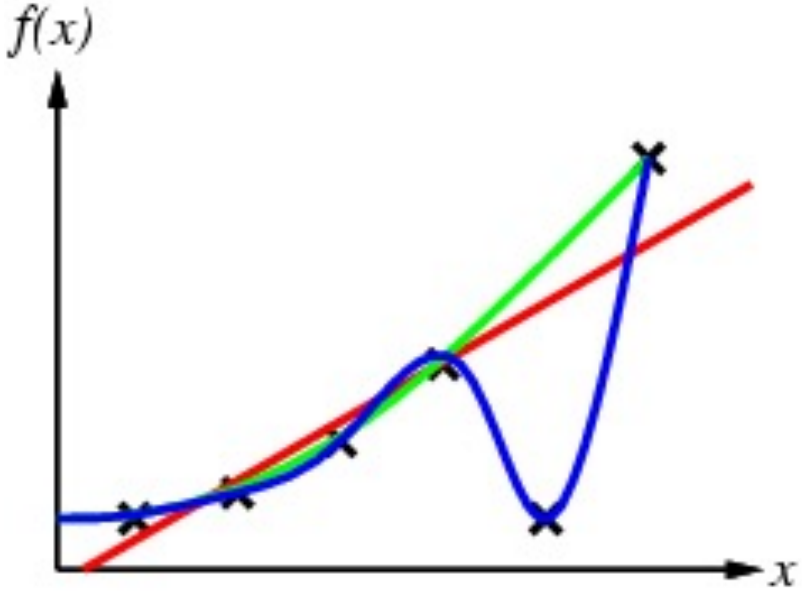
Training the Model

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



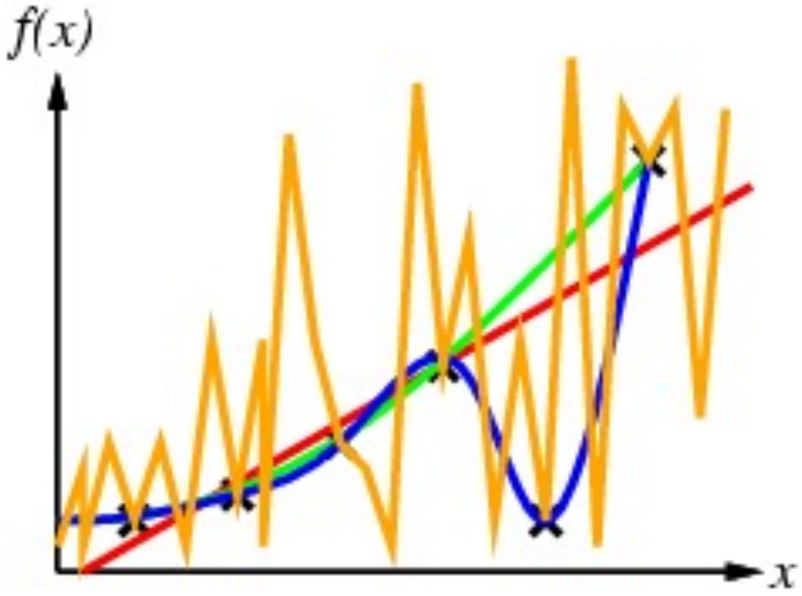
Training the Model

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



Training the Model

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:

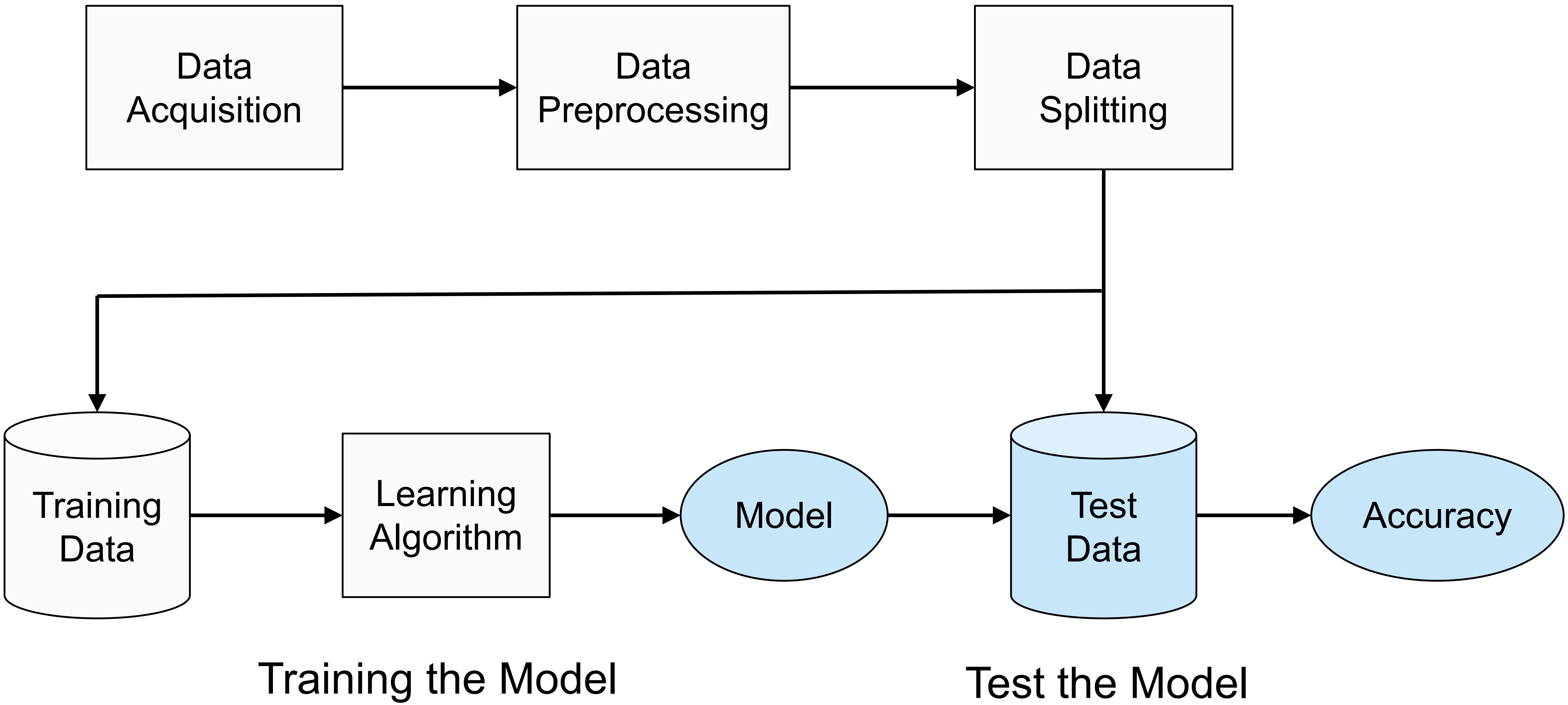


Ockham's razor: prefer the simplest hypothesis consistent with data

Training the Model

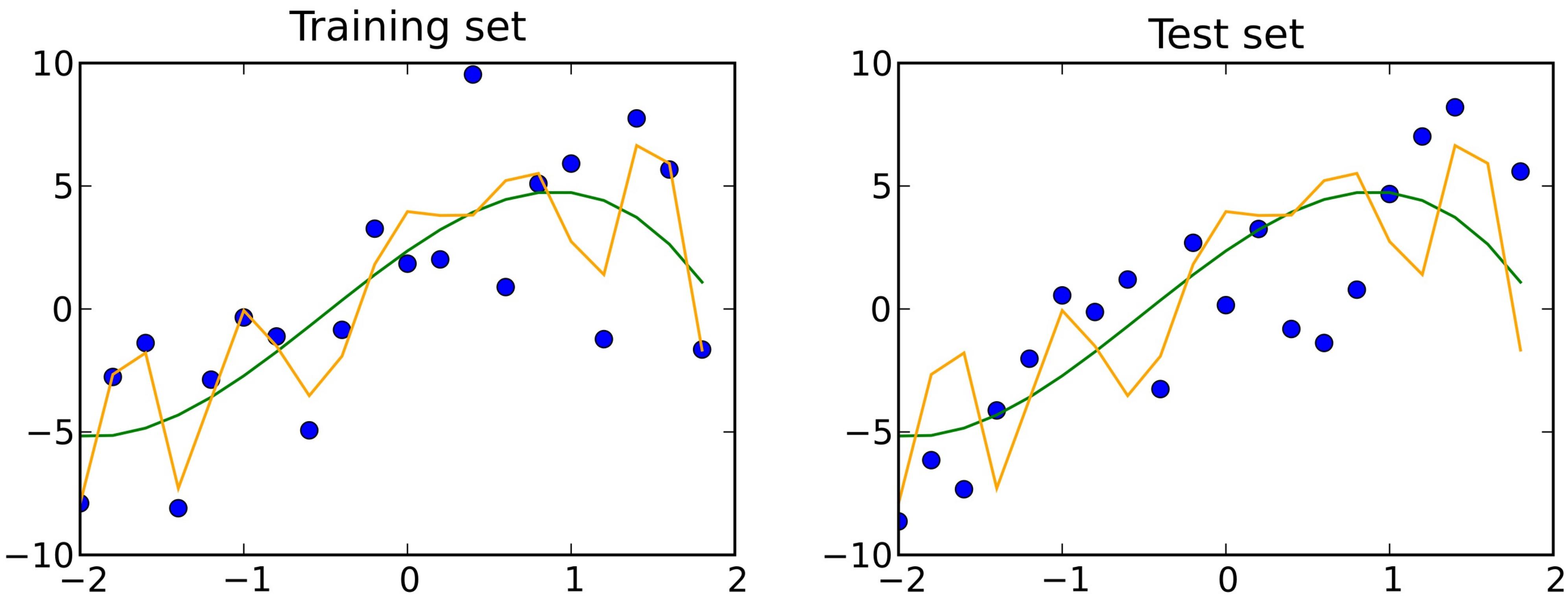
- Hypotheses must **generalize** to correctly classify/predict instances not in the training data.
- Simply memorizing training examples is a consistent hypothesis that does not generalize.
- *Occam's razor*:
  - Finding a *simple* hypothesis helps ensure generalization.

Supervised Learning Process



Testing the Model

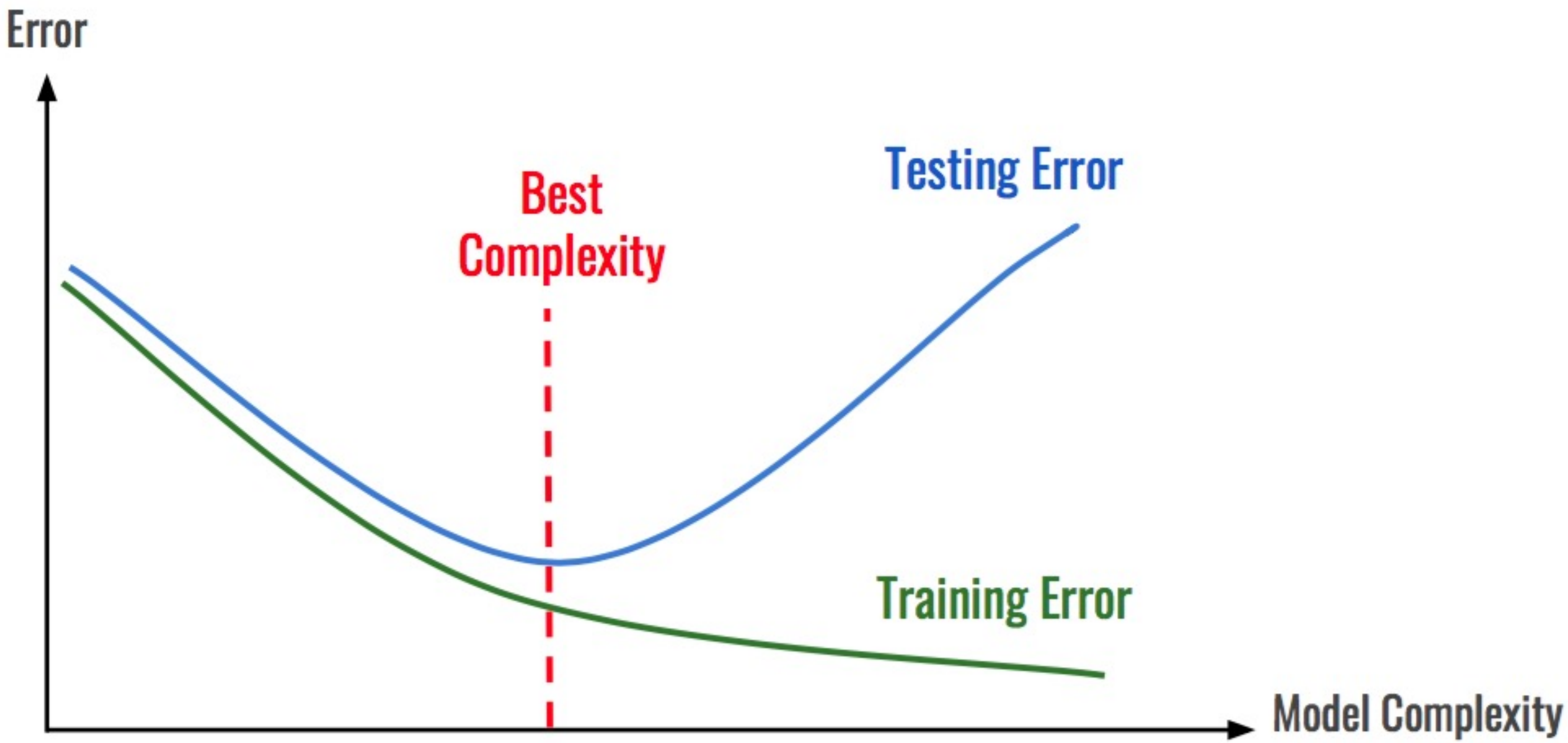
- Test the model using unseen test data to assess the model accuracy
- Avoid overfitting at the learning stage



Wikipedia

Testing the Model

- Test the model using unseen test data to assess the model accuracy
- Avoid overfitting at the learning stage



Cohort Problem CS5

CS5. *Standardization:* Write a function that takes in data frame where all the column are the features and normalize each column according to the following formula.

$$normalized = \frac{data - \mu}{\sigma}$$

Cohort Problem CS6

CS5. *Splitting Data Randomly:* Create a function to split the Data Frame randomly. The function should have the following arguments:

- df_feature: which is the data frame for the features.
- df_target: which is the data frame for the target.
- random_state: which is the seed used to split randomly.
- test_size: which is the fraction for the test data set (0 to 1), by default is set to 0.5

Thank You!