Machine Learning

Outline

I. ML and applications

II. Supervised learning

Com S 474/574 *Introduction to Machine Learning* (Spring 2021)

* A large portion of the material is drawn from Dr. Jin Tian's notes.

** Figures are from either the textbook site or Dr. Jin Tian's notes.

AI and Machine Learning

- AI is the enterprise of design and analysis of intelligent agents.
- Intelligent behavior requires knowledge (e.g., model of the environment).
- Explicit specifications of the knowledge needed for specific tasks are hard, and often infeasible.
- How to acquire knowledge?

AI and Machine Learning

- AI is the enterprise of design and analysis of intelligent agents.
- Intelligent behavior requires knowledge (e.g., model of the environment).
- Explicit specifications of the knowledge needed for specific tasks are hard, and often infeasible.
- How to acquire knowledge?

Machine learning (ML) refers to the process in which a computer

- observes some data,
- builds a model based on the data, and
- uses the model as both a hypothesis about the world and a piece of problem solving software.

Learning Agents

- Learning modifies the agent's decision mechanisms to improve performance.

    - Which component is to be improved.
    - Which prior knowledge the agent has, which influences the model.
    - What data and feedback on that data is available.

- Environment changes over time – learning needs to adapt to changes.

- Learning is essential for unknown environments.

Applications of ML

- Agriculture
- Anatomy
- Adaptive websites
- Affective computing
- Banking
- Bioinformatics
- Brain–machine interfaces
- Cheminformatics
- Citizen science
- Computer networks
- Computer vision
- Credit-card fraud detection
- Data quality
- DNA sequence classification
- Economics
- Financial market analysis[75]
- General game playing
- Handwriting recognition
- Information retrieval
- Insurance
- Internet fraud detection
- Linguistics
- Machine learning control
- Machine perception
- Machine translation
- Marketing
- Medical diagnosis
- Natural language processing
- Natural language understanding
- Online advertising
- Optimization
- Recommender systems
- Robot locomotion
- Search engines
- Sentiment analysis
- Sequence mining
- Software engineering
- Speech recognition
- Structural health monitoring
- Syntactic pattern recognition
- Telecommunication
- Theorem proving
- Time series forecasting
- User behavior analytics

* From Wikipedia: https://en.wikipedia.org/wiki/Machine_learning#Applications

Applications of ML

- Agriculture
- Anatomy
- Adaptive websites
- Affective computing
- Banking
- Bioinformatics
- Brain–machine interfaces
- Cheminformatics
- Citizen science
- Computer networks
- Computer vision
- Credit-card fraud detection
- Data quality
- DNA sequence classification
- Economics
- Financial market analysis[75]
- General game playing
- Handwriting recognition
- Information retrieval
- Insurance
- Internet fraud detection
- Linguistics
- Machine learning control
- Machine perception
- Machine translation
- Marketing
- Medical diagnosis
- Natural language processing
- Natural language understanding
- Online advertising
- Optimization
- Recommender systems
- Robot locomotion
- Search engines
- Sentiment analysis
- Sequence mining
- Software engineering
- Speech recognition
- Structural health monitoring
- Syntactic pattern recognition
- Telecommunication
- Theorem proving
- Time series forecasting
- User behavior analytics

* From Wikipedia: https://en.wikipedia.org/wiki/Machine_learning#Applications

Data Mining

- Huge amounts of data are available from

  science, medicine, economics, geography, environment, sports, …

- Data is a potentially valuable resource.

- Raw data are useless – need techniques to automatically extract information from it.

  - Data: recorded facts

  - Information: patterns underlying the data

- Machine learning techniques automatically find patterns in data.

The Game-Weather Problem

Weather condition for playing a certain game:

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	Normal	False	Yes
...	...	...	...	...

The Game-Weather Problem

Weather condition for playing a certain game:

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	Normal	False	Yes
...	...	...	...	...

Learned classification rules:

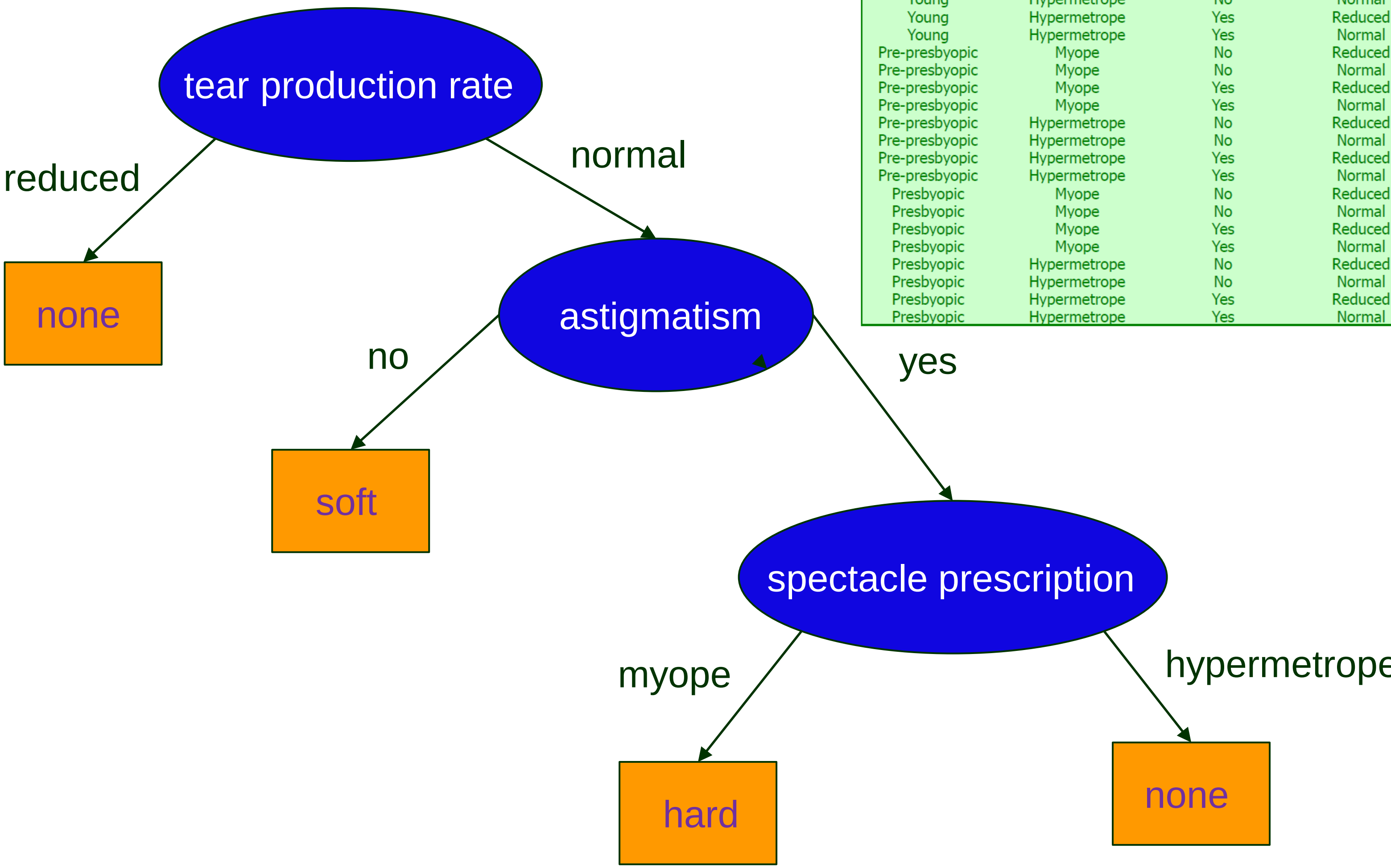
```
If outlook = sunny and humidity = high then play = no
If outlook = rainy and windy = true then play = no
If outlook = overcast then play = yes
If humidity = normal then play = yes
If none of the above then play = yes
```

Contact Lense Data

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	No	Reduced	None
Young	Myope	No	Normal	Soft
Young	Myope	Yes	Reduced	None
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	No	Reduced	None
Young	Hypermetrope	No	Normal	Soft
Young	Hypermetrope	Yes	Reduced	None
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	No	Reduced	None
Pre-presbyopic	Myope	No	Normal	Soft
Pre-presbyopic	Myope	Yes	Reduced	None
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	No	Reduced	None
Pre-presbyopic	Hypermetrope	No	Normal	Soft
Pre-presbyopic	Hypermetrope	Yes	Reduced	None
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	No	Reduced	None
Presbyopic	Myope	No	Normal	None
Presbyopic	Myope	Yes	Reduced	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	No	Reduced	None
Presbyopic	Hypermetrope	No	Normal	Soft
Presbyopic	Hypermetrope	Yes	Reduced	None
Presbyopic	Hypermetrope	Yes	Normal	None

Learned Decision Tree

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	No	Reduced	None
Young	Myope	No	Normal	Soft
Young	Myope	Yes	Reduced	None
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	No	Reduced	None
Young	Hypermetrope	No	Normal	Soft
Young	Hypermetrope	Yes	Reduced	None
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	No	Reduced	None
Pre-presbyopic	Myope	No	Normal	Soft
Pre-presbyopic	Myope	Yes	Reduced	None
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	No	Reduced	None
Pre-presbyopic	Hypermetrope	No	Normal	Soft
Pre-presbyopic	Hypermetrope	Yes	Reduced	None
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	No	Reduced	None
Presbyopic	Myope	No	Normal	None
Presbyopic	Myope	Yes	Reduced	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	No	Reduced	None
Presbyopic	Hypermetrope	No	Normal	Soft
Presbyopic	Hypermetrope	Yes	Reduced	None
Presbyopic	Hypermetrope	Yes	Normal	None



Predicting CPU Performance

209 different computer configurations

	Cycle time (ns)	Main memory (Kb)		Cache (Kb)	Channels		Performance
	MYCT	MMIN	MMAX	CACH	CHMIN	CHMAX	PRP
1	125	256	6000	256	16	128	198
2	29	8000	32000	32	8	32	269
…							
208	480	512	8000	32	0	0	67
209	480	1000	4000	0	0	0	45

Predicting CPU Performance

209 different computer configurations

	Cycle time (ns)	Main memory (Kb)		Cache (Kb)	Channels		Performance
	MYCT	MMIN	MMAX	CACH	CHMIN	CHMAX	PRP
1	125	256	6000	256	16	128	198
2	29	8000	32000	32	8	32	269
...							
208	480	512	8000	32	0	0	67
209	480	1000	4000	0	0	0	45

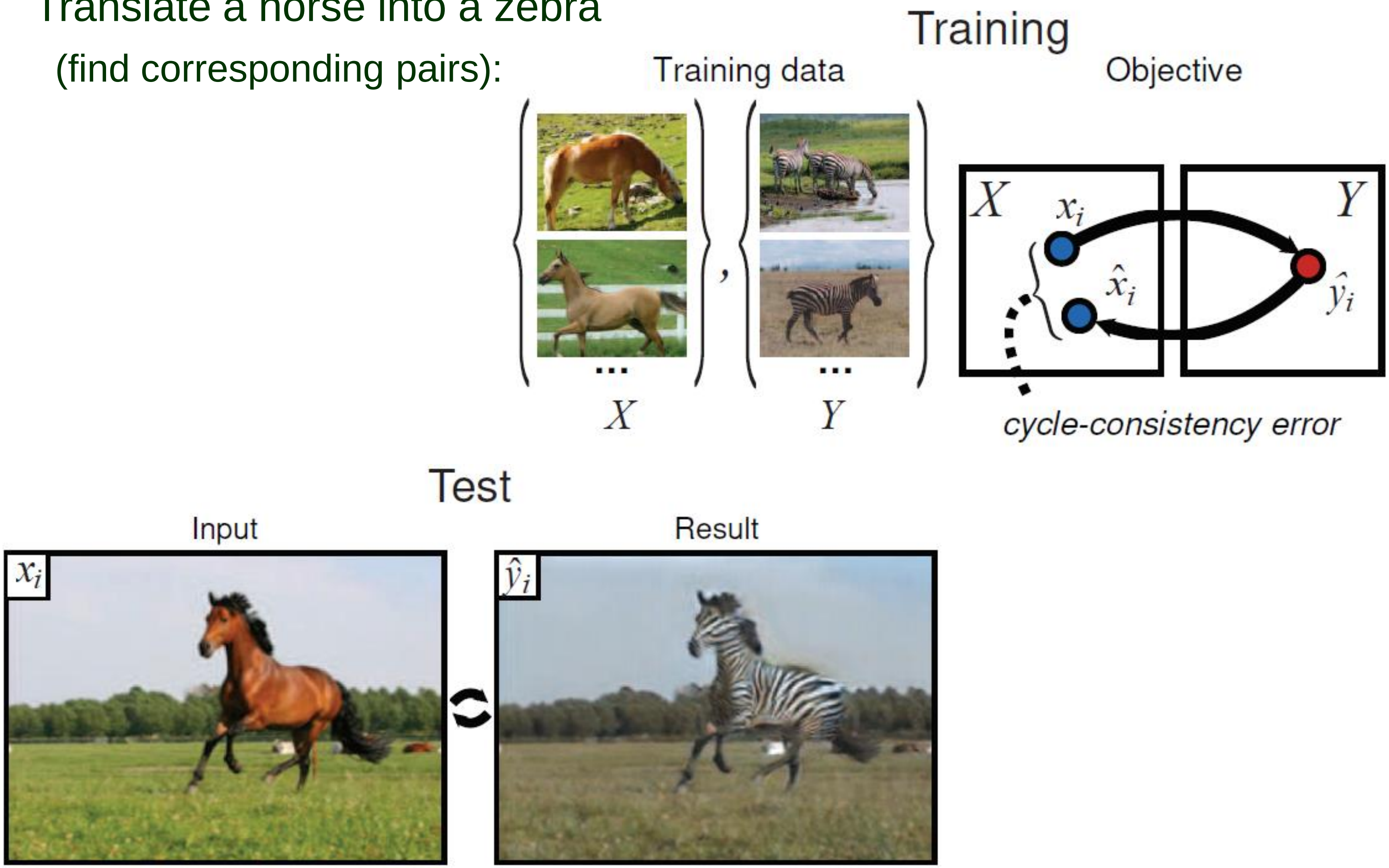
Function obtained through linear regression (fitting):

```
PRP = -55.9 + 0.0489 MYCT + 0.0153 MMIN + 0.0056 MMAX
      + 0.6410 CACH - 0.2700 CHMIN + 1.480 CHMAX
```

Image Translation

Translate a horse into a zebra

(find corresponding pairs):



Supervised Learning

The agent observes input-output pairs and learns a function that maps from input to output.

Supervised Learning

The agent observes input-output pairs and learns a function that maps from input to output.

Problem Given a *training set* of N input-output pairs:

$$(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$$

where each pair was generated by an unknown function $y = f(x)$, discover a function h to approximate f.

Supervised Learning

The agent observes input-output pairs and learns a function that maps from input to output.

Problem Given a *training set* of N input-output pairs:

$$(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$$

where each pair was generated by an unknown function $y = f(x)$, discover a function h to approximate f.

hypothesis (or *model*) drawn from a *hypothesis space* $\mathcal{H}$ of possible functions

Supervised Learning

The agent observes input-output pairs and learns a function that maps from input to output.

Problem Given a *training set* of N input-output pairs:

$$(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$$

where each pair was generated by an unknown function $y = f(x)$, discover a function h to approximate f.

hypothesis (or *model*) drawn from a *hypothesis space* $\mathcal{H}$ of possible functions

chosen according to some prior knowledge about data generation

Supervised Learning

The agent observes input-output pairs and learns a function that maps from input to output.

Problem Given a *training set* of N input-output pairs:

$$(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$$

where each pair was generated by an unknown function $y = f(x)$, discover a function h to approximate f.

hypothesis (or *model*) drawn from a *hypothesis space* $\mathcal{H}$ of possible functions

chosen according to some prior knowledge about data generation

$y_1, y_2, \dots, y_n$: *ground truth* to be predicted by our model

Best-Fit Function

- With noise in data, we cannot expect an exact match with the ground truth, namely, $h(x_i) = y_i$, for $1 \leq i \leq N$.

Best-Fit Function

- ♣ With noise in data, we cannot expect an exact match with the ground truth, namely, $h(x_i) = y_i$, for $1 \leq i \leq N$.
- ♣ Instead, we look for a best-fit function h for which each $h(x_i)$ is close y_i.

Best-Fit Function

- With noise in data, we cannot expect an exact match with the ground truth, namely, $h(x_i) = y_i$, for $1 \leq i \leq N$.

- Instead, we look for a best-fit function h for which each $h(x_i)$ is close y_i.

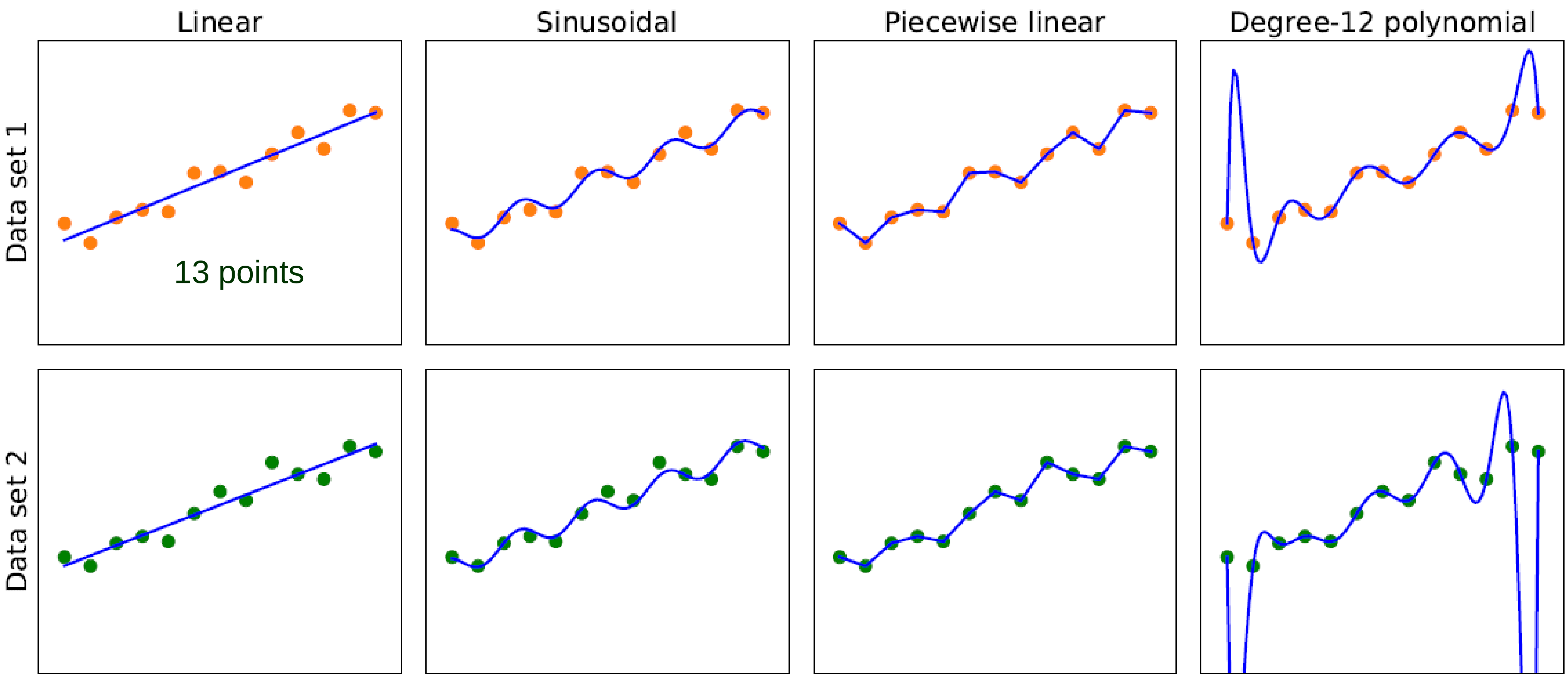
- The true measure of h is how it handles inputs it has not seen.

  Test set: a second sample of (x_i, y_i) pairs

- h generalizes well if it matches the test set with high accuracy.

Fitting

(Least-squares fitting: http://web.cs.iastate.edu/~cs577/handouts/data-fit.pdf)



$h(x) = w_1 x + w_0$

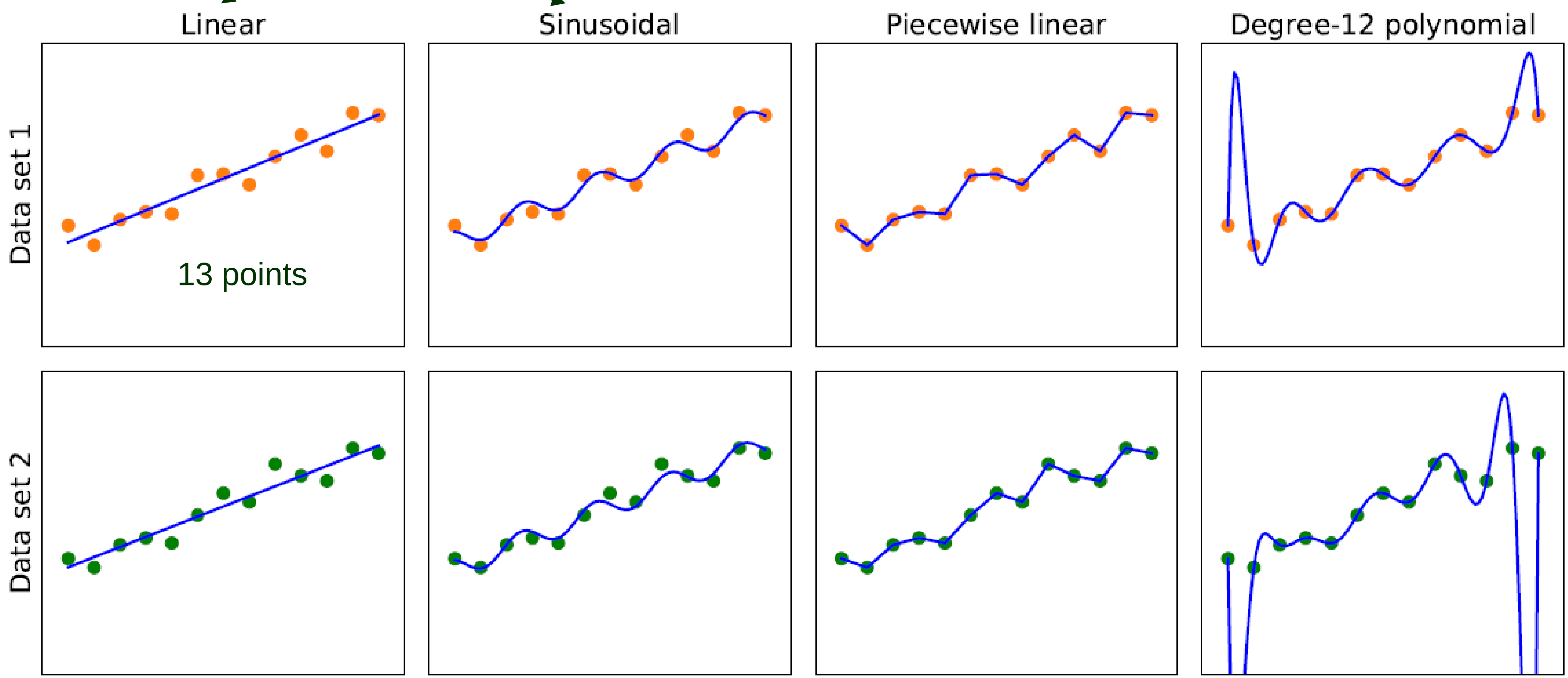
$h(x) = w_1 x + \sin(w_0 x)$

$h_i(x) = w_{i1} x + w_{i0}$, $1 \leq i \leq 12$ such that $h_j(x_{j+1}) = h_{j+1}(x_{j+1})$ for $1 \leq j \leq 11$.

$h(x) = \sum_{i=0}^{12} w_i x^i$

Fitting

(Least-squares fitting: http://web.cs.iastate.edu/~cs577/handouts/data-fit.pdf)



$h(x) = w_1 x + w_0$

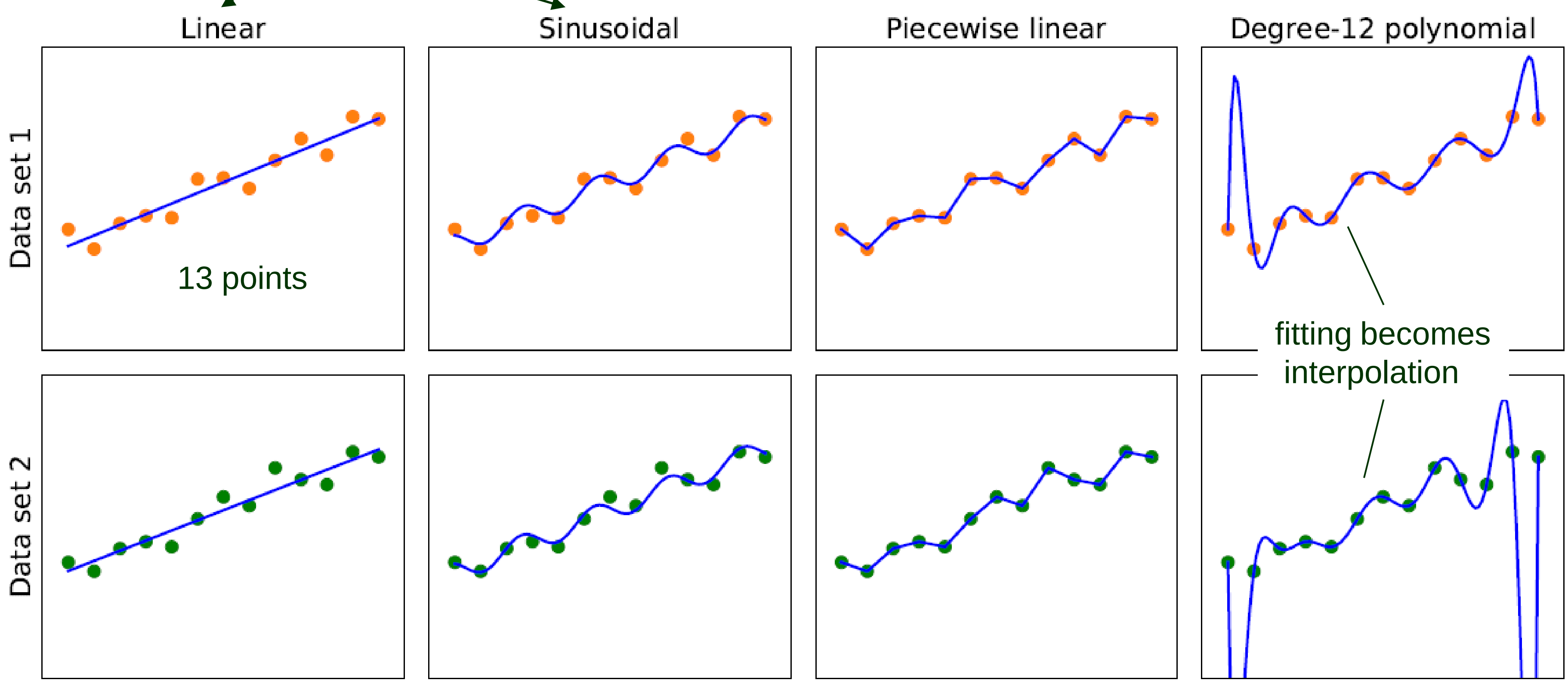
$h(x) = w_1 x + \sin(w_0 x)$

$h_i(x) = w_{i1} x + w_{i0}$, $1 \le i \le 12$ such that $h_j(x_{j+1}) = h_{j+1}(x_{j+1})$ for $1 \le j \le 11$.

$h(x) = \sum_{i=0}^{12} w_i x^i$

Fitting

(Least-squares fitting: http://web.cs.iastate.edu/~cs577/handouts/data-fit.pdf)



$h(x) = w_1 x + w_0$

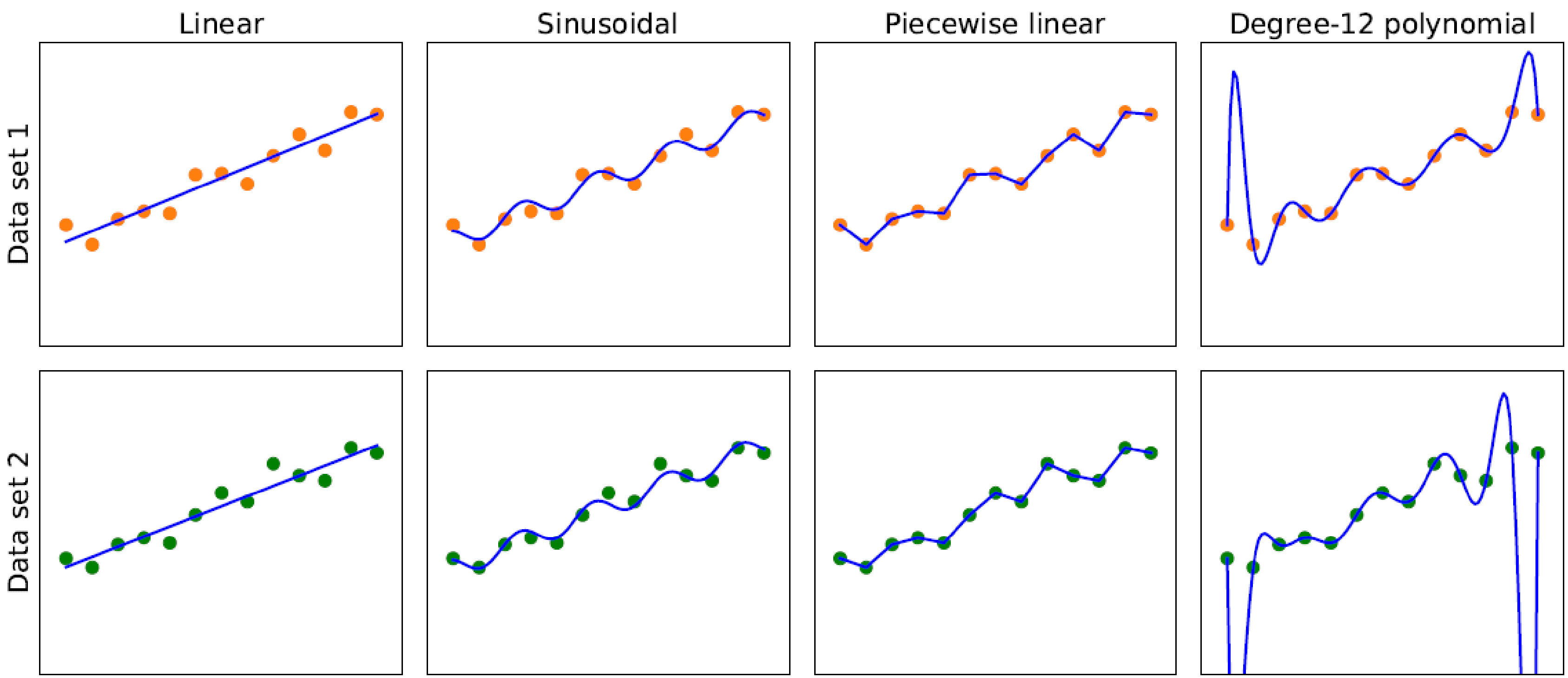
$h(x) = w_1 x + \sin(w_0 x)$

$h_i(x) = w_{i1} x + w_{i0}$, $1 \le i \le 12$ such that $h_j(x_{j+1}) = h_{j+1}(x_{j+1})$ for $1 \le j \le 11$.

$h(x) = \sum_{i=0}^{12} w_i x^i$

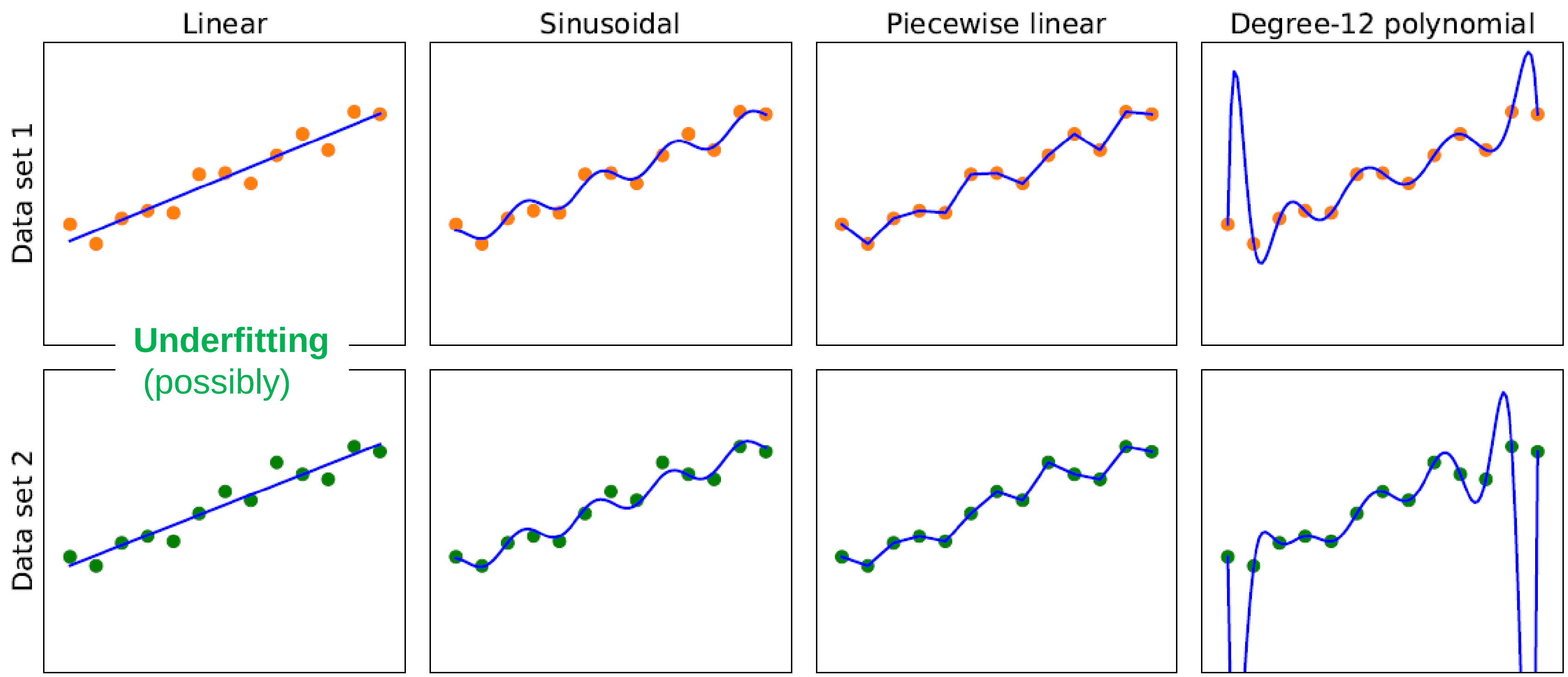
Underfitting, Variance & Overfitting

- A hypothesis is *underfitting* when it fails to find a pattern in the data.
    - Such a hypothesis has low bias.



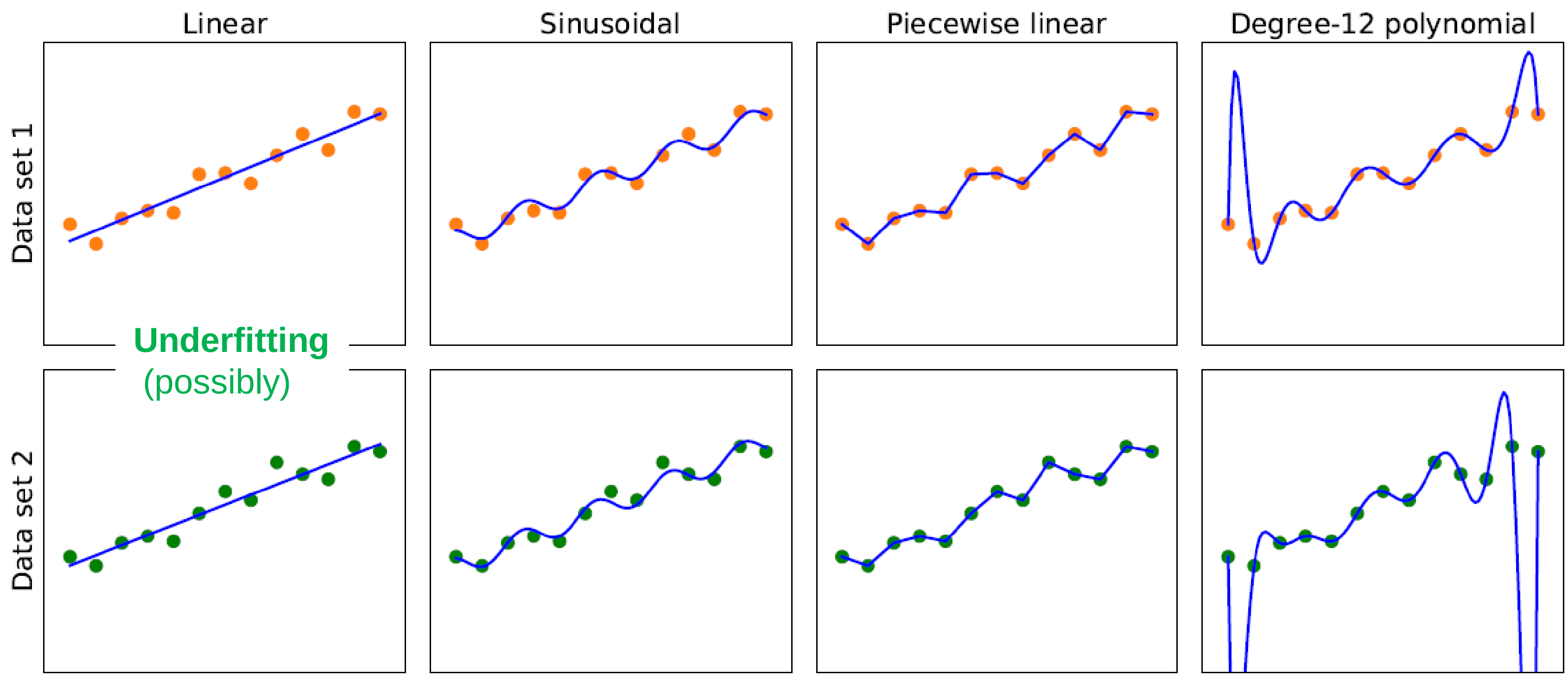
Underfitting, Variance & Overfitting

- A hypothesis is *underfitting* when it fails to find a pattern in the data.
  - Such a hypothesis has low bias.



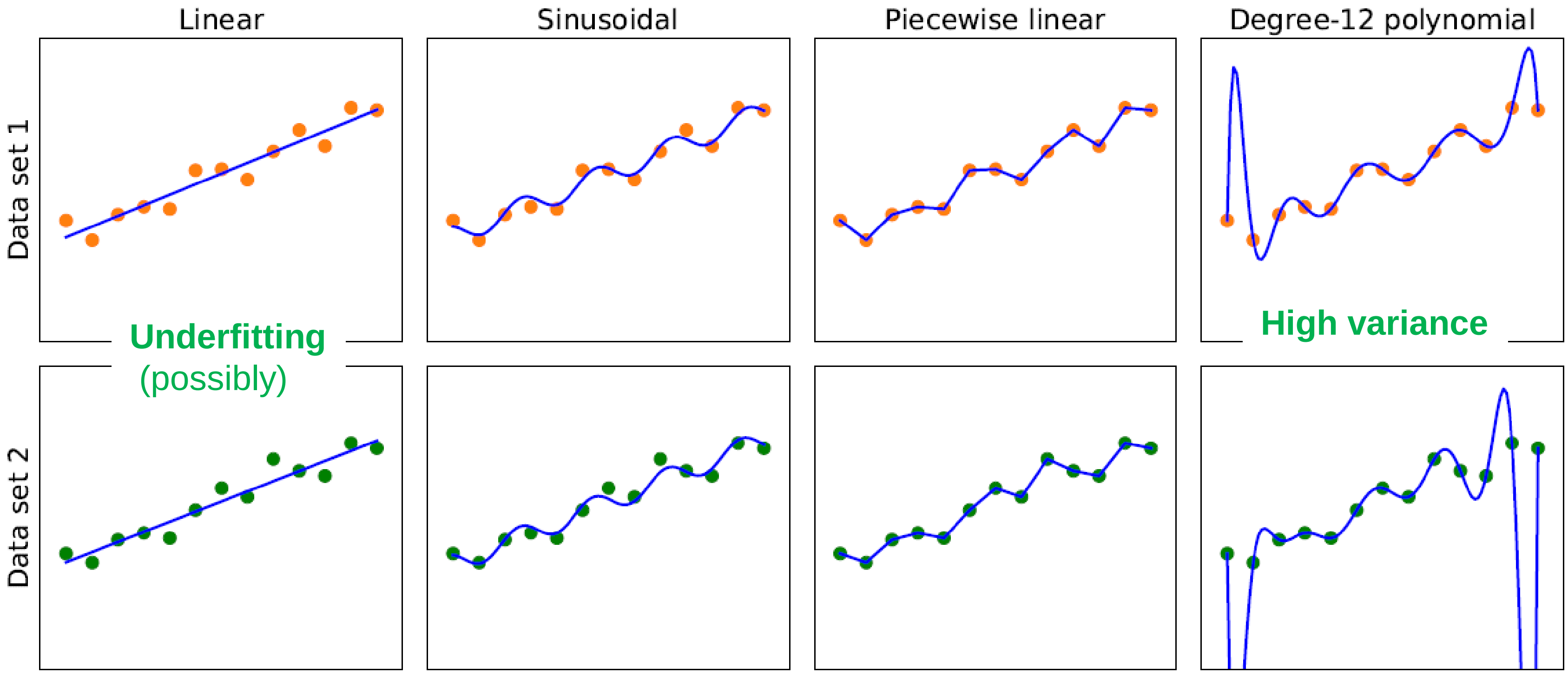
Underfitting, Variance & Overfitting

- A hypothesis is *underfitting* when it fails to find a pattern in the data.
  - Such a hypothesis has low bias.
- *Variance* characterizes the amount of change in the hypothesis due to fluctuation in the training data.



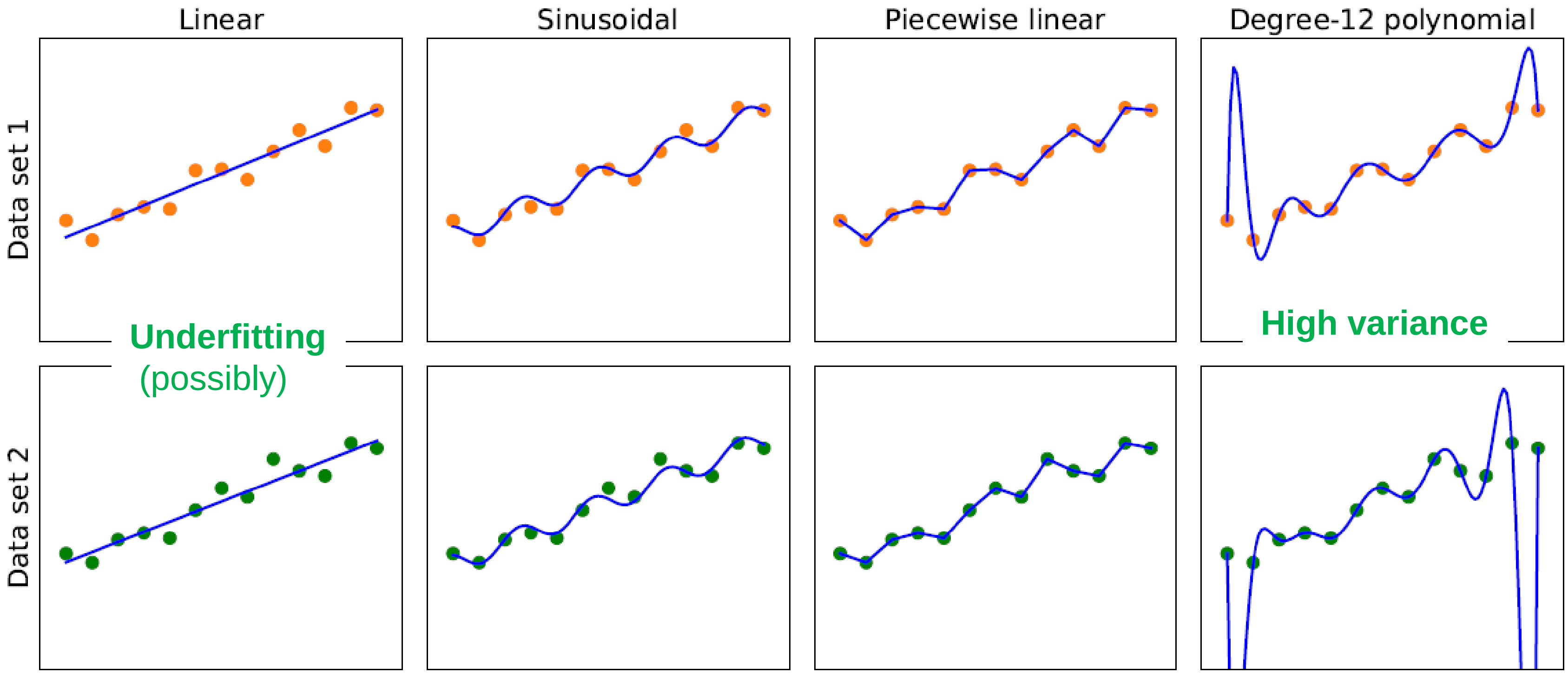
Underfitting, Variance & Overfitting

- A hypothesis is *underfitting* when it fails to find a pattern in the data.
  - Such a hypothesis has low bias.
- *Variance* characterizes the amount of change in the hypothesis due to fluctuation in the training data.



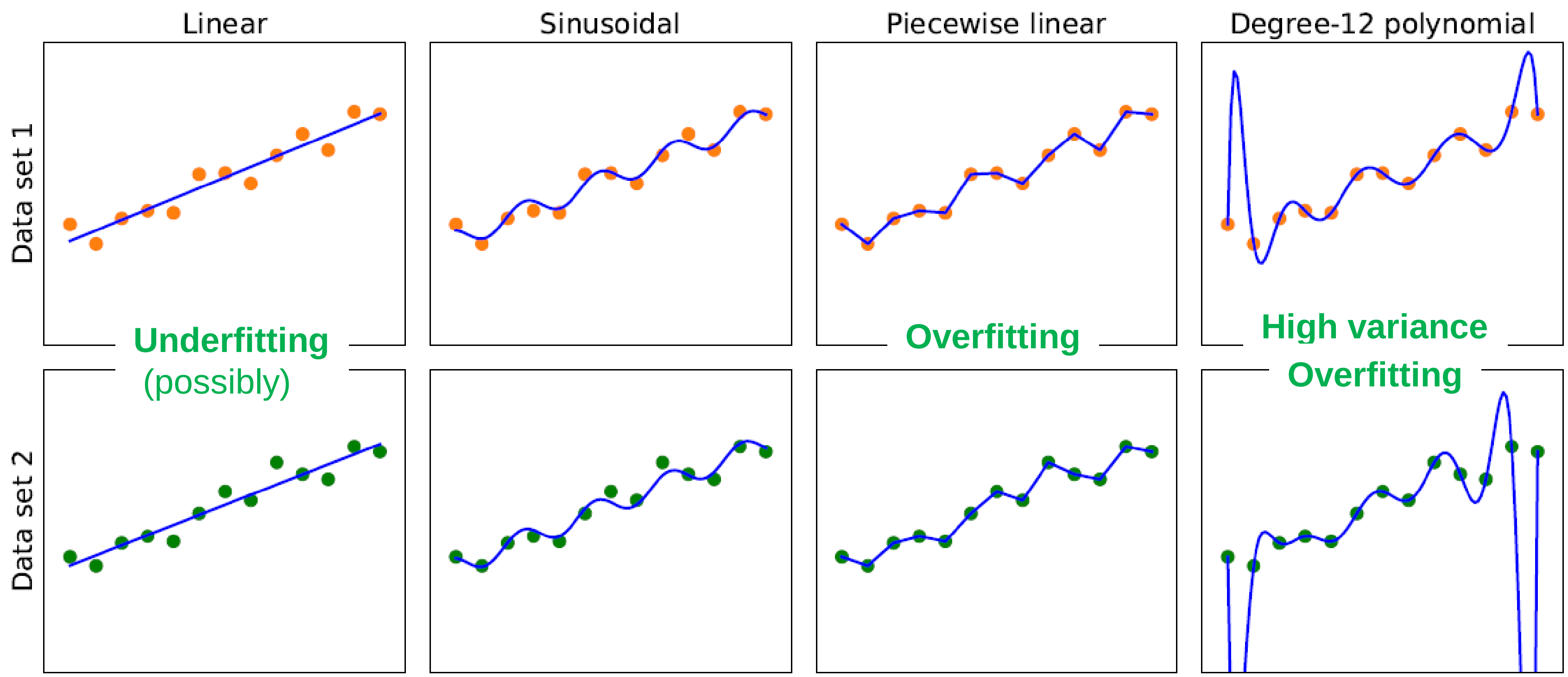
Underfitting, Variance & Overfitting

- A hypothesis is *underfitting* when it fails to find a pattern in the data.
  - Such a hypothesis has low bias.
- *Variance* characterizes the amount of change in the hypothesis due to fluctuation in the training data.
- A hypothesis is *overfitting* when it pays too much attention to the particular training set.



Underfitting, Variance & Overfitting

- A hypothesis is *underfitting* when it fails to find a pattern in the data.
  - Such a hypothesis has low bias.
- *Variance* characterizes the amount of change in the hypothesis due to fluctuation in the training data.
- A hypothesis is *overfitting* when it pays too much attention to the particular training set.



Best Hypothesis

It depends on what we knew was represented by the data, or, what we were expecting from the data ...

Best Hypothesis

It depends on what we knew was represented by the data, or, what we were expecting from the data ...

Choosing the hypothesis that is *most probable* given the data:

Best Hypothesis

It depends on what we knew was represented by the data, or, what we were expecting from the data ...

Choosing the hypothesis that is *most probable* given the data:

$$\boxed{h^*} = \operatorname*{arcmax}_{h \in \mathcal{H}} P(h \mid data)$$

Best Hypothesis

It depends on what we knew was represented by the data, or, what we were expecting from the data ...

Choosing the hypothesis that is *most probable* given the data:

$$h^* = \underset{h \in \mathcal{H}}{\operatorname{arcmax}} P(h \mid data)$$

$$= \underset{h \in \mathcal{H}}{\operatorname{arcmax}} P(data \mid h) P(h) \qquad \text{(Bayes' rule)}$$

Best Hypothesis

It depends on what we knew was represented by the data, or, what we were expecting from the data ...

Choosing the hypothesis that is *most probable* given the data:

$$h^* = \operatorname*{arcmax}_{h \in \mathcal{H}} P(h \mid data)$$

$$= \operatorname*{arcmax}_{h \in \mathcal{H}} P(data \mid h) P(h) \qquad \text{(Bayes' rule)}$$

A simple hypothesis space $\mathcal{H}$ is often preferred:

- ♠ The more expressiveness of $\mathcal{H}$, the higher the computational cost of finding a good hypothesis within that space.
- ♣ We will likely be using h for evaluations after we have learned it.

Restaurant Waiting

Problem Decide whether to wait for a table at a restaurant.

Restaurant Waiting

Problem Decide whether to wait for a table at a restaurant.

Output: a Boolean variable *WillWait* (true where we do wait for a table).

Restaurant Waiting

Problem Decide whether to wait for a table at a restaurant.

Output: a Boolean variable *WillWait* (true where we do wait for a table).

Input: a vector of ten attributes, each with discrete values:

Restaurant Waiting

Problem Decide whether to wait for a table at a restaurant.

Output: a Boolean variable *WillWait* (true where we do wait for a table).

Input: a vector of ten attributes, each with discrete values:

1. *Alternate*: whether there is a suitable alternative restaurant nearby.
2. *Bar*: whether the restaurant has a comfortable bar area to wait in.
3. *Fri/Sat*: true on Fridays and Saturdays.
4. *Hungry*: whether we are hungry right now.
5. *Patrons*: how many people are in the restaurant (values: *None*, *Some*, and *Full*).
6. *Price*: the restaurant's price range ($, $$, $$$).
7. *Raining*: whether it is raining outside.
8. *Reservation*: whether we made a reservation.
9. *Type*: the kind of restaurant (*French*, *Italian*, *Thai*, or *burger*).
10. *WaitEstimate*: host's wait estimate: 0-10, 10-30, 30-60, or >60 minutes.

Restaurant Waiting

Problem Decide whether to wait for a table at a restaurant.

Output: a Boolean variable *WillWait* (true where we do wait for a table).

Input: a vector of ten attributes, each with discrete values:

1. *Alternate*: whether there is a suitable alternative restaurant nearby.
2. *Bar*: whether the restaurant has a comfortable bar area to wait in.
3. *Fri/Sat*: true on Fridays and Saturdays.
4. *Hungry*: whether we are hungry right now.
5. *Patrons*: how many people are in the restaurant (values: *None*, *Some*, and *Full*).
6. *Price*: the restaurant's price range ($, $$, $$$).
7. *Raining*: whether it is raining outside.
8. *Reservation*: whether we made a reservation.
9. *Type*: the kind of restaurant (*French*, *Italian*, *Thai*, or *burger*).
10. *WaitEstimate*: host's wait estimate: 0-10, 10-30, 30-60, or >60 minutes.

$2^6 \times 3^2 \times 4^2 = 9{,}216$ possible combinations of attribute values.

Training Examples

Example	Input Attributes										Output
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	$y_1 = Yes$
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	$y_2 = No$
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	$y_3 = Yes$
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10–30*	$y_4 = Yes$
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	$y_5 = No$
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	$y_6 = Yes$
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	$y_7 = No$
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	$y_8 = Yes$
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	$y_9 = No$
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	$y_{10} = No$
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	$y_{11} = No$
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	$y_{12} = Yes$

- The correct output is given for only 12 out of 9,216 examples.

Training Examples

Example	Input Attributes										Output
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	$y_1 = Yes$
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	$y_2 = No$
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	$y_3 = Yes$
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10–30*	$y_4 = Yes$
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	$y_5 = No$
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	$y_6 = Yes$
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	$y_7 = No$
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	$y_8 = Yes$
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	$y_9 = No$
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	$y_{10} = No$
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	$y_{11} = No$
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	$y_{12} = Yes$

- The correct output is given for only 12 out of 9,216 examples.
- We need to make our best guess at the missing 9,204 output values.