Intro to ML

March 10, 2020
Data Science CSCI 1951A
Brown University

Instructor: Ellie Pavlick
HTAs: Josh Levin, Diane Mutako, Sol Zitter

Announcements

- This class is going viral! (Funny? No? Too soon?)
  - Not officially, but starting to prep just in case
  - Trial run on Thursday
  - Quizzes and Clickers will remain both valid until further notice
- Questions?

Today

- ML “preliminaries”—terminology, basic building blocks, conceptual background
- The two faces of linear regression
- Training with Stochastic Gradient Descent

Today

- **ML “preliminaries”—terminology, basic building blocks, conceptual background**

- The two faces of linear regression

- Training with Stochastic Gradient Descent

Quick Clicker Q!

How much ML experience have you had?

(a) None at all. I have obviously heard of ML but I’ve never really dealt with it.

(b) Small amount of informal experience. I’ve read articles/blog posts and gotten the gist of how it works.

(c) Like (b), but I’ve followed along an coded some models myself

(d) Comfortable. I’ve taken an ML class.

(e) Very comfortable. I’ve taken an ML class/classes and I’ve built models myself for research projects or internships.

Quick Clicker Q!

Characterize your knowledge of ML:

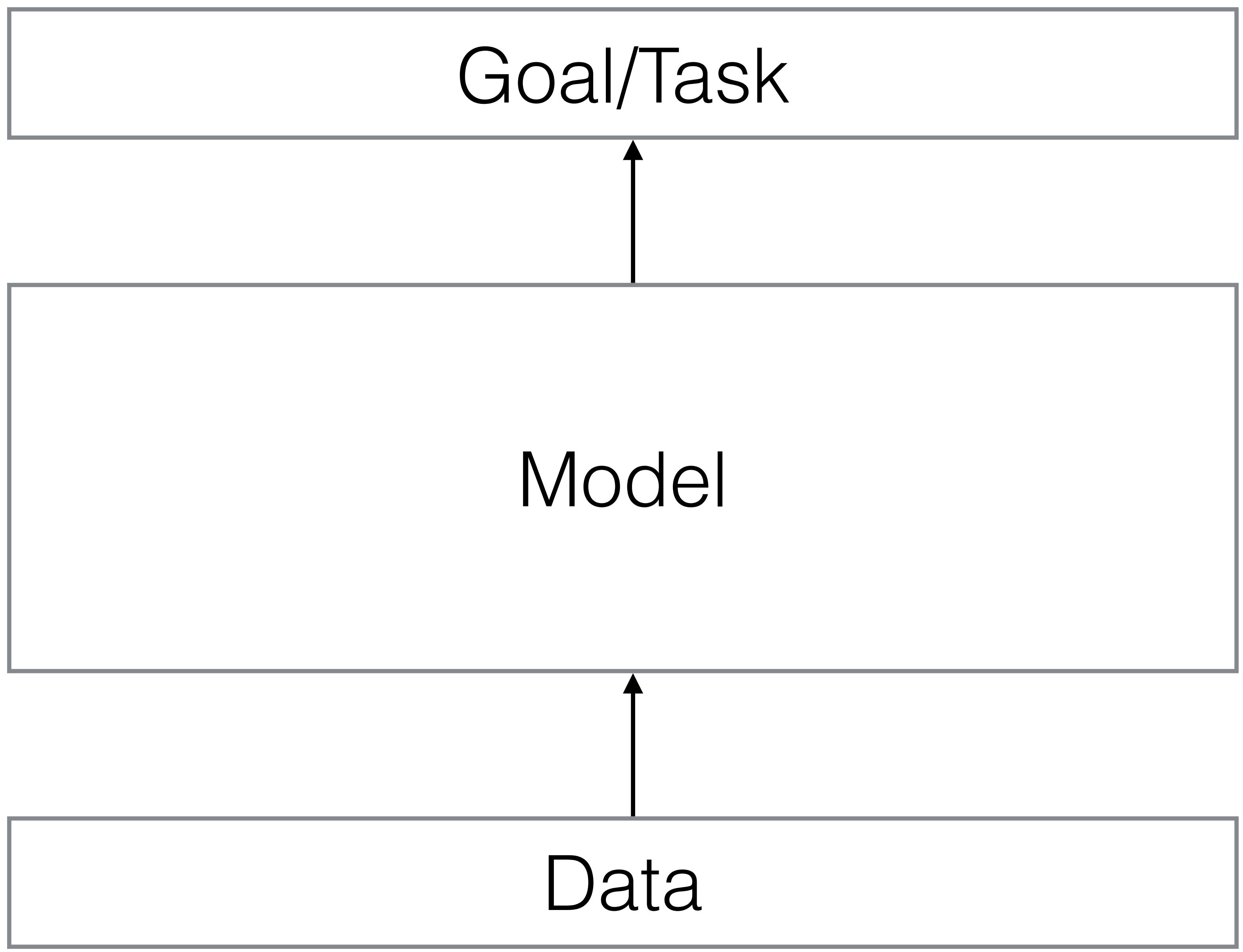
(a) Mostly “conventional” ML

(b) Mostly deep learning

(c) Equally comfortable with both

(d) Not comfortable with either

Oversimplified ML



Oversimplified ML

Goal/Task

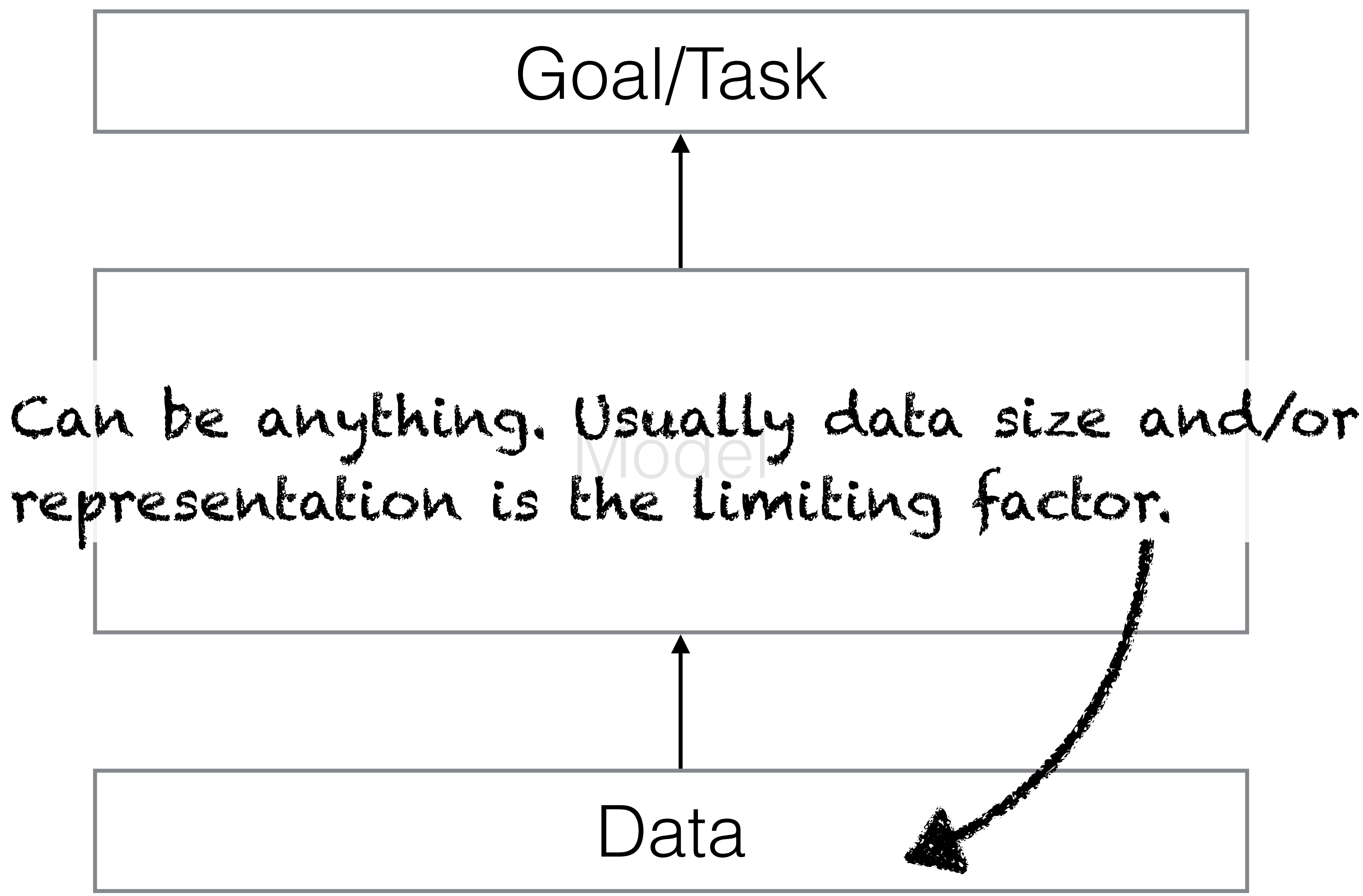
Model

Data

Prediction of some kind, e.g.:

- price of a stock (number)
- sentiment of a piece of text (discrete label)
- objects in an image (tagging)
- strategy for a video game (sequence)
- parse tree of a sentence (tree structure)

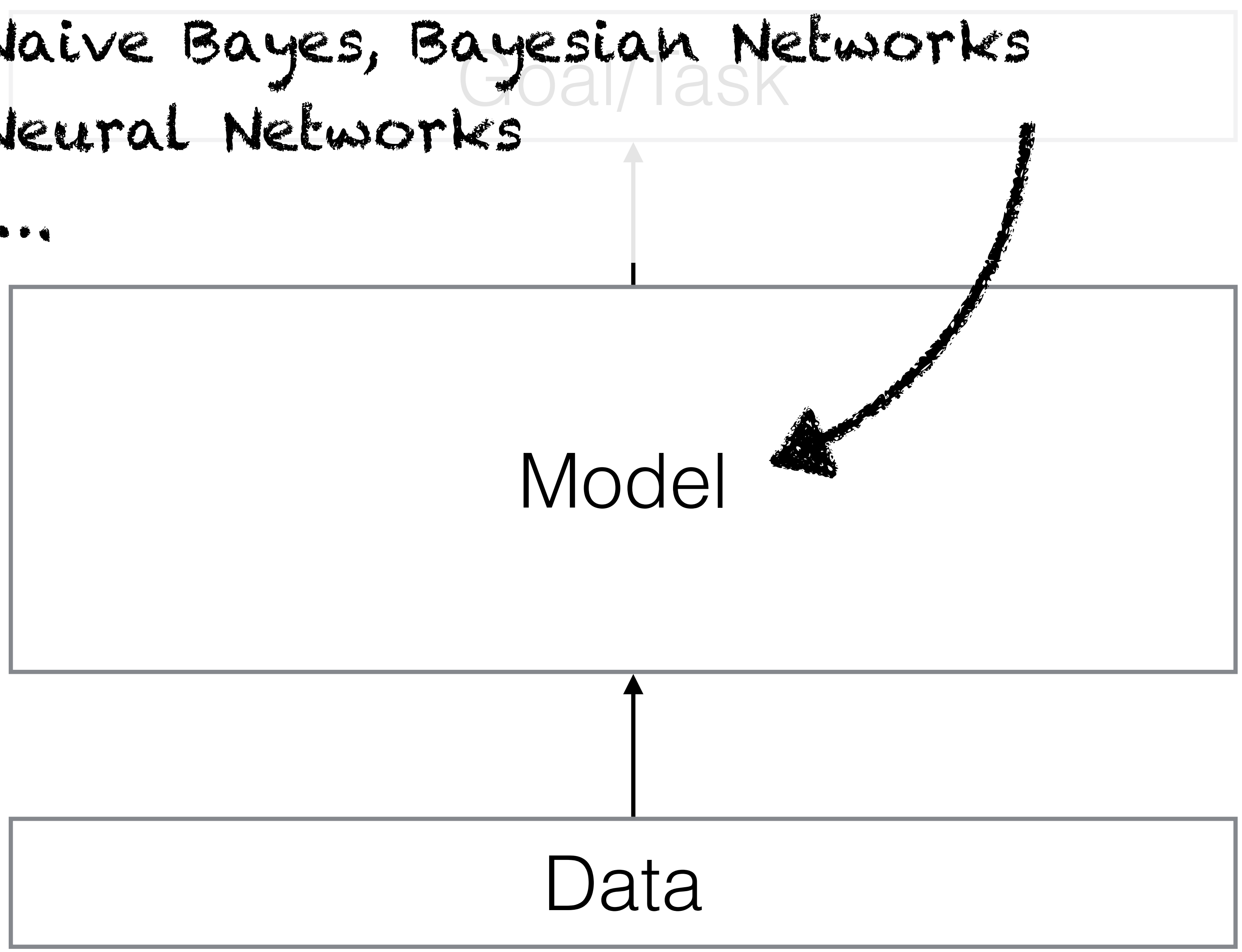
Oversimplified ML



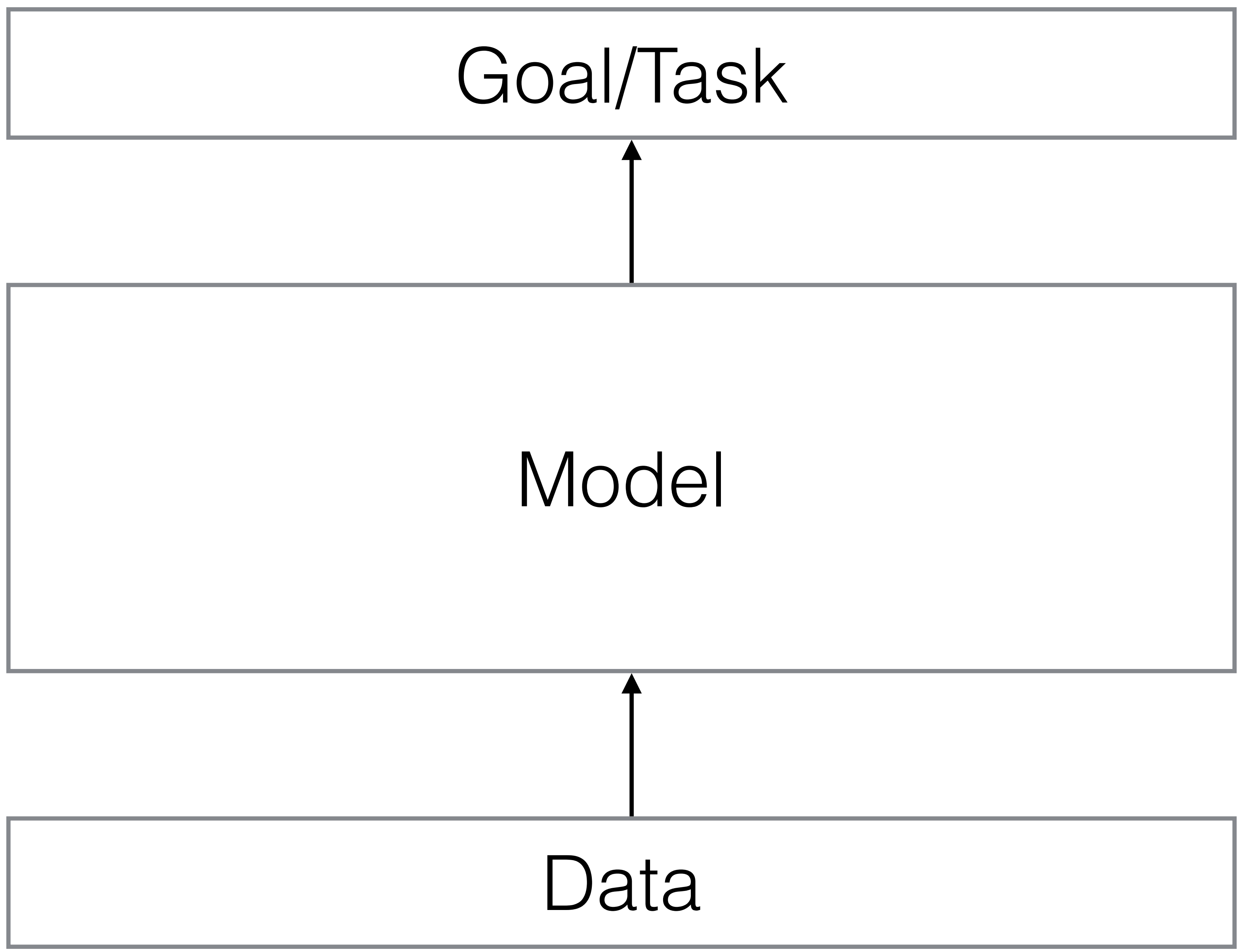
Oversimplified ML

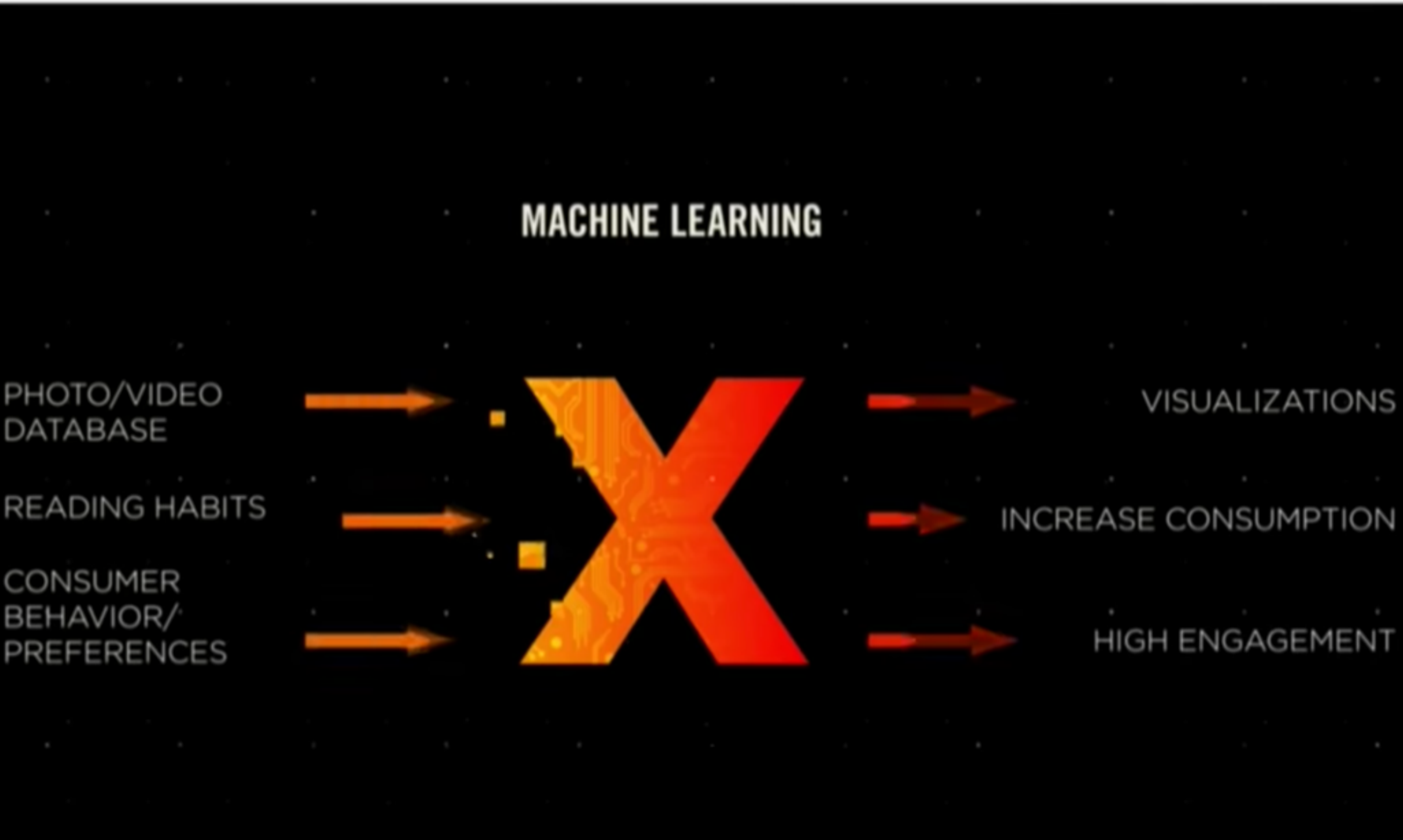
Decisions about how the problem is structured AND how to estimate parameters

- linear/logistic regression
- SVMs
- Naive Bayes, Bayesian Networks
- Neural Networks
- ....

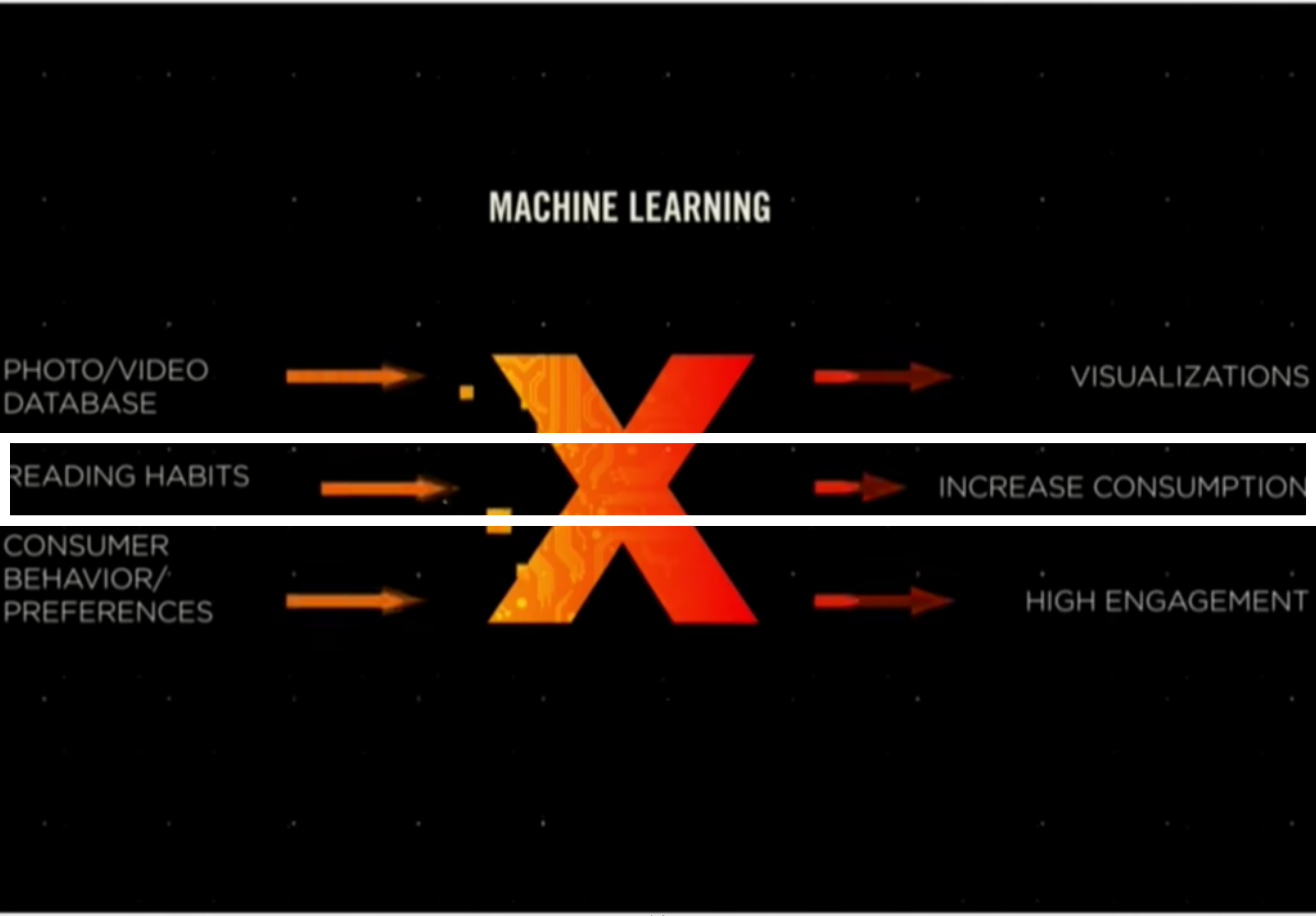


Defining an ML problem

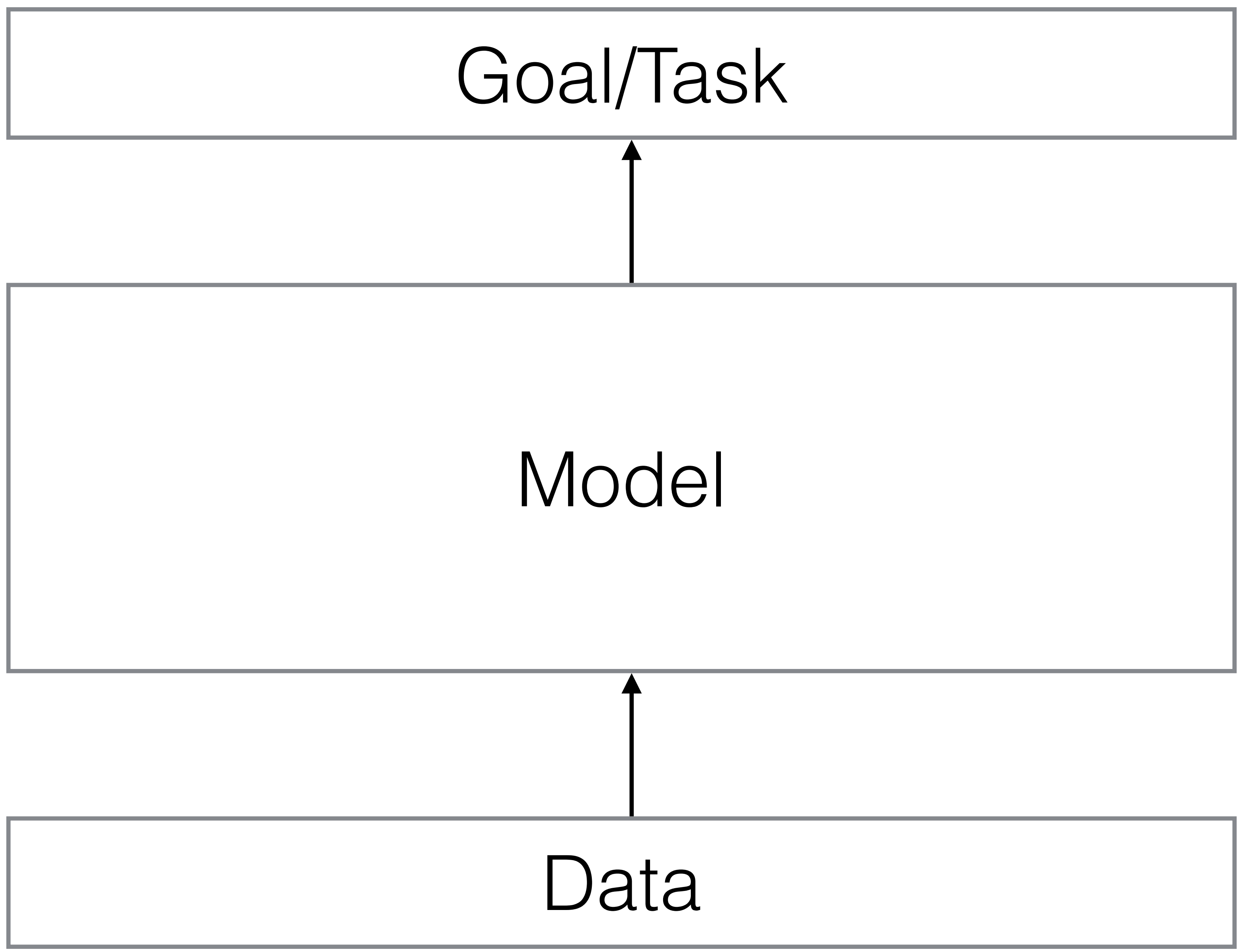




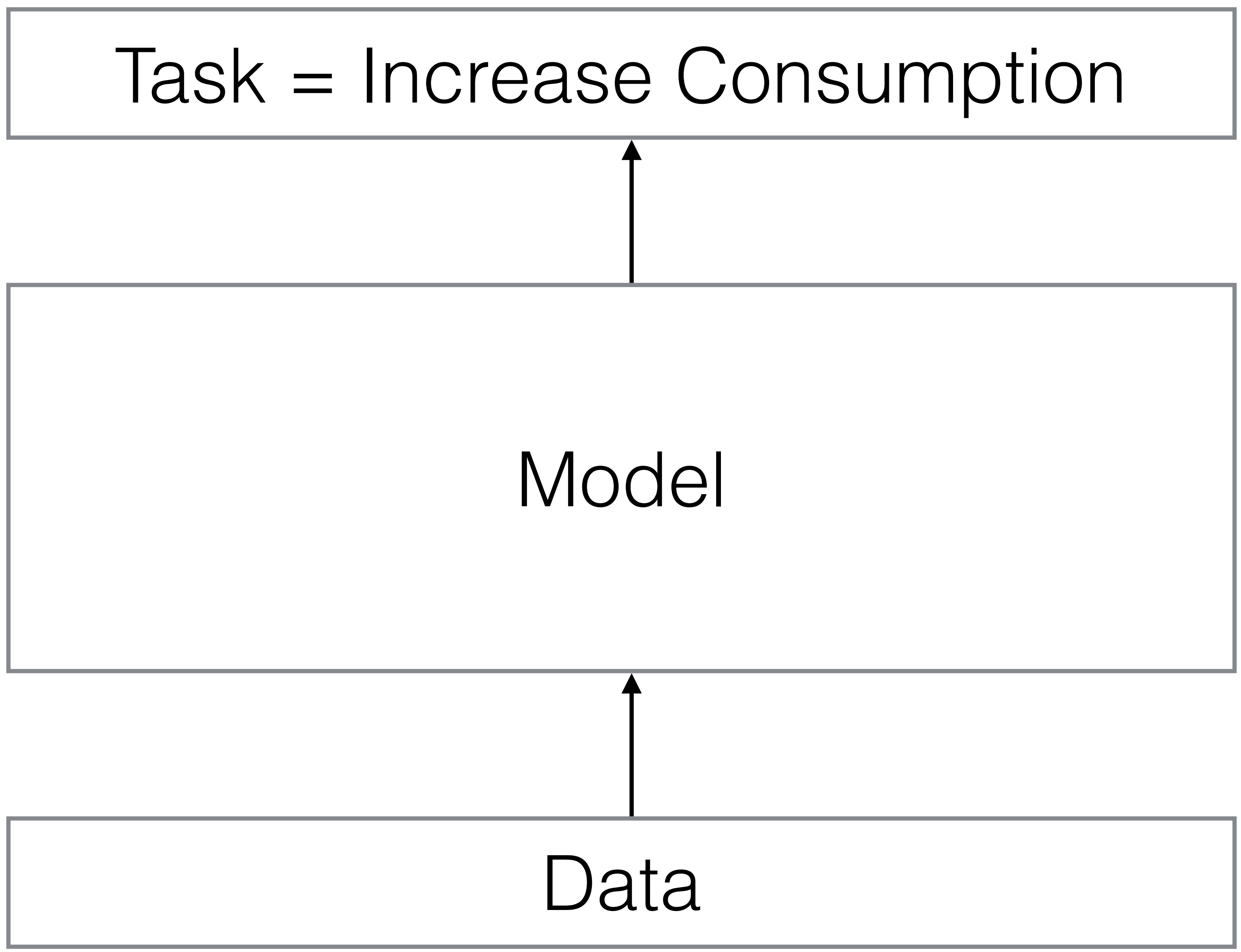
https://youtu.be/bq2_wSsDwkQ?t=682


MACHINE LEARNING
PHOTO/VIDEO DATABASE
READING HABITS
CONSUMER BEHAVIOR/ PREFERENCES
VISUALIZATIONS
INCREASE CONSUMPTION
HIGH ENGAGEMENT

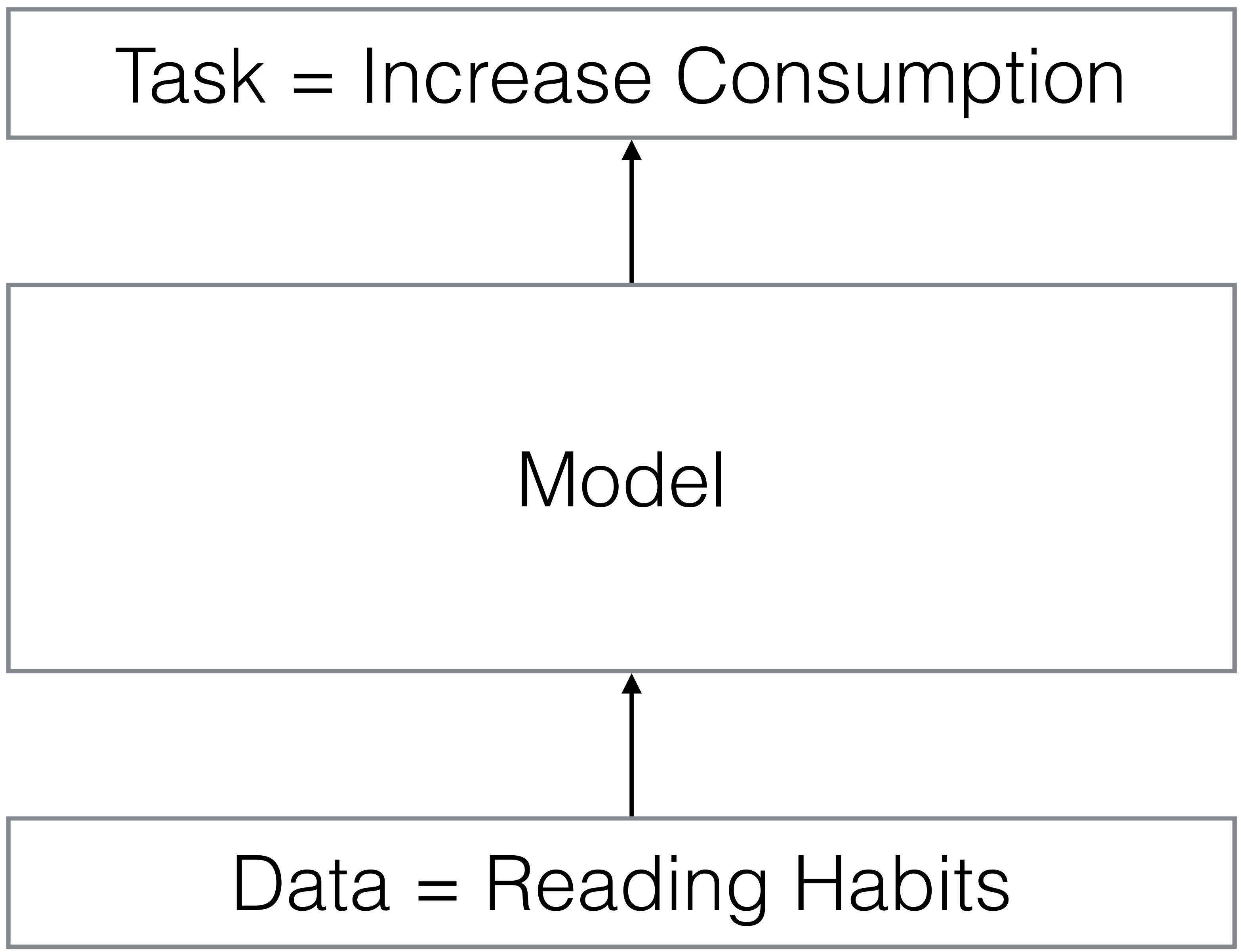
Defining an ML problem



Defining an ML problem



Defining an ML problem



Defining an ML problem

Task = Increase Consumption

↑

Model

↑

Data = Reading Habits

Defining an ML problem

- What is “machine learnable”?

Defining an ML problem

- What is “machine learnable”?
- ~~Like…basically everything, right?~~

Defining an ML problem

- What is “machine learnable”?
- ~~Like…basically everything, right?~~ WRONG!!

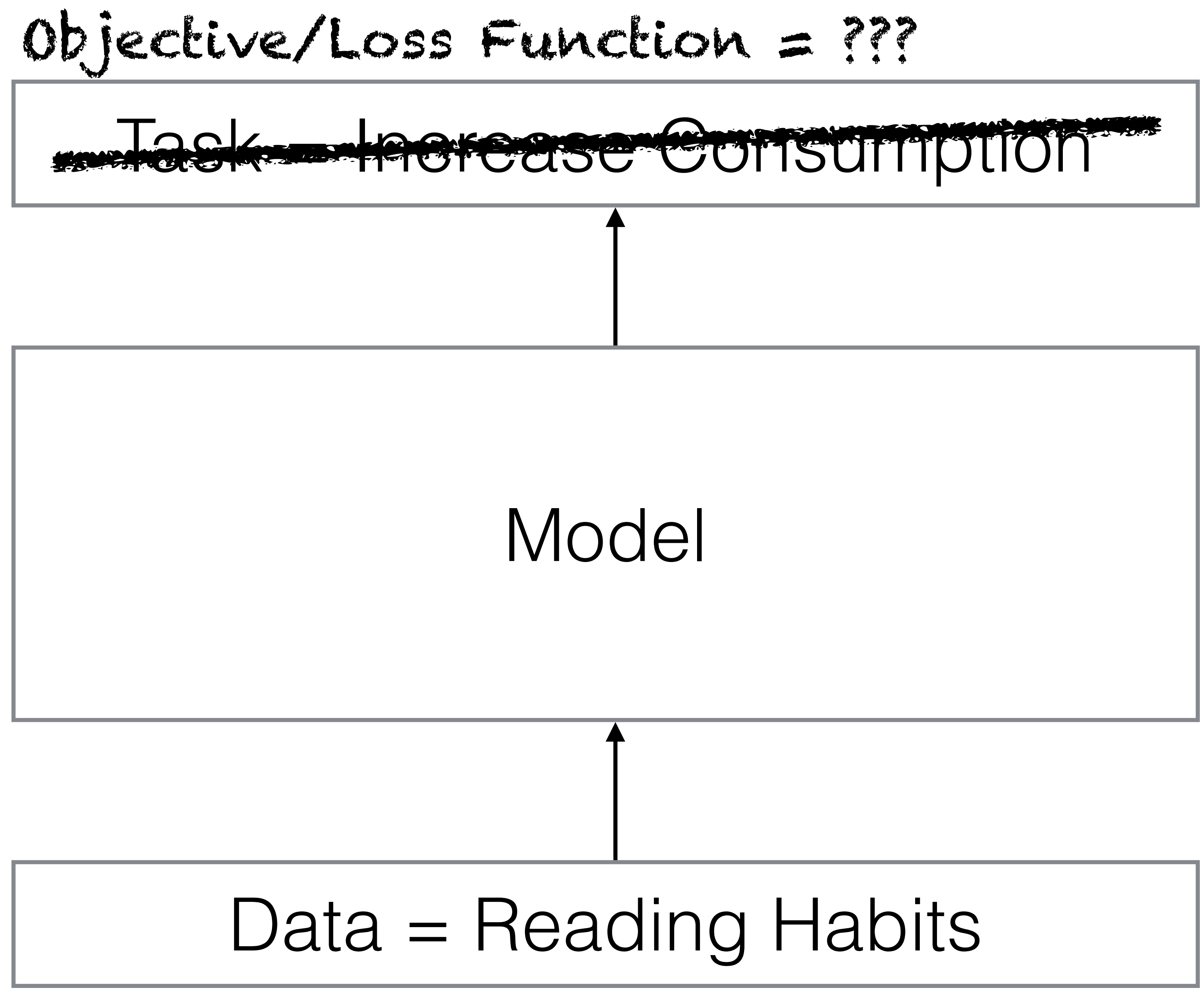
Defining an ML problem

- What is "machine learnable"?
- ~~Like…basically everything, right?~~ WRONG!! (kind of)

Defining an ML problem

- What is “machine learnable”?
- ~~Like…basically everything, right?~~ WRONG!! (kind of)
- Input features need to be concrete and representable. Definition of “success” needs to be quantifiable (and, right now, usually differentiable).

Defining an ML problem



Defining an ML problem

Objective/Loss Function = ???

~~Task = Increase Consumption~~

↑

Model

↑

~~Data = Reading Habits~~

Features = ???

Defining an ML problem

Objective/Loss Function = ???

~~Task = Increase Consumption~~

Model

~~Data = Reading Habits~~

Features = ???

Prediction Target

- Goal = Increase consumption of “content” NOS for your ~~clickbait farm~~ pulitzer-prize worthy publication

Prediction Target

- Goal = Increase consumption of “content” NOS for your ~~clickbait farm~~ pulitzer-prize worthy publication
- Prediction target….ideas?

Clicker Question!

Prediction Target

- Goal = Increase consumption of “content” NOS for your ~~clickbait farm~~ pulitzer-prize worthy publication

- Prediction target….ideas?

> - Time spent on site (avg. per user/total)
> - Number of users
> - Number of articles read (need to define “read”)
> - Number of articles clicked on
> - Time per article
> - Articles shared…

Prediction Target

- Goal = Increase consumption of “content” NOS for your ~~clickbait farm~~ pulitzer-prize worthy publication

- Prediction target….ideas?

> - Time spent on site (avg. per user/total)
> - Number of users
> - Number of articles read (need to define “read”)
> - **Number of articles clicked on**
> - Time per article
> - Articles shared…

Clicker Question!

Prediction Target

- Goal = Increase consumption of “content” NOS for your ~~clickbait farm~~ pulitzer-prize worthy publication

- Prediction target….ideas?

- Objective/Loss Function…ideas?

- Time spent on site (avg. per user/total)
- Number of users
- Number of articles read (need to define “read”)
- **Number of articles clicked on**
- Time per article
- Articles shared…

Prediction Target

- Goal = Increase consumption of “content” NOS for your ~~clickbait farm~~ pulitzer-prize worthy publication

- Prediction ta

- Objective/Lo

- Time s
- Numbe
- Number of articles read (need to define “read”)
- **Number of articles clicked on**
- Time per article
- Articles shared…

- Difference between predicted and true value
- Squared difference between predicted and true value
- Predicted probability of true value
- Whether you were right or wrong (binary)

Prediction Target

- Goal = Increase consumption of “content” NOS for your ~~clickbait farm~~ pulitzer-prize worthy publication
- Prediction ta
- Objective/Lo

- Difference between predicted and true value
- **Squared difference between predicted and true value**
- Predicted probability of true value
- Whether you were right or wrong (binary)

- Time s
- Numbe
- Number of articles read (need to define “read”)
- **Number of articles clicked on**
- Time per article
- Articles shared…

Defining an ML problem

Objective/Loss Function = ???

~~Task = Increase Consumption~~

Model

~~Data = Reading Habits~~

Features = ???

Defining an ML problem

Objective/Loss Function = squared difference between predicted total number of clicks and actual total number of clicks

~~Task = Increase Consumption~~

Model

~~Data = Reading Habits~~

Features = ???

Defining an ML problem

Objective/Loss Function = squared difference between predicted total number of clicks and actual total number of clicks

~~Task = Increase Consumption~~

Model

~~Data = Reading Habits~~

Features = ???

Features

- Data = Reading habits collected via ~~unauthorized ever-present cookies and remote control of webcam~~ user-consented GDPR-compliant data usage agreements

Features

- Data = Reading habits collected via ~~unauthorized ever-present cookies and remote control of webcam~~ user-consented GDPR-compliant data usage agreements

- Features….ideas?

Clicker Question!

Features

- Data = Reading habits collected via ~~unauthorized ever-present cookies and remote control of webcam~~ user-consented GDPR-compliant data usage agreements

- Features….ideas?

Features

- Data = Reading habits collected via ~~unauthorized ever-present cookies and remote control of webcam~~ user-consented GDPR-compliant data usage agreements

- Features

  - Article topic
  - Recency (minutes since release)
  - Words in title/snippet
  - Presence of photo
  - Reading level
  - Fonts/layouts
  - User location
  - Topics of articles the user has read previously
  - Number of likes

  …

Features

- Data = Reading habits collected via ~~unauthorized ever-present cookies and remote control of webcam~~ user-consented GDPR-compliant data usage agreements

- Features

  - Article topic
  - **Recency** (minutes since release)
  - **Words in title**/snippet
  - **Presence of photo**
  - **Reading level**
  - Fonts/layouts
  - User location
  - Topics of articles the user has read previously
  - Number of likes

  …

Features

- Recency: Float
- Words in title: String
- Presence of photo: Boolean
- Reading level: Integer

Features

Clicks	Recency	Reading Level	Photo	Title
10	1.3	11	1	"New Tax Guidelines"
1000	1.7	3	1	"This 600lb baby…"
1000000	2.4	2	1	"18 reasons you should *never* look at this cat unless you…"
1	5.9	19	0	"The Brothers Karamazov: a neo-post-globalist perspective"

Features

y

Clicks	Recency	Reading Level	Photo	Title
10	1.3	11	1	“New Tax Guidelines”
1000	1.7	3	1	“This 600lb baby…”
1000000	2.4	2	1	“18 reasons you should *never* look at this cat unless you…”
1	5.9	19	0	“The Brothers Karamazov: a neo-post-globalist perspective”

Features

Clicks	Recency	Reading Level	Photo	Title
10	1.3	11	1	“New Tax Guidelines”
1000	1.7	3	1	“This 600lb baby…”
1000000	2.4	2	1	“18 reasons you should *never* look at this cat unless you…”
1	5.9	19	0	“The Brothers Karamazov: a neo-post-globalist perspective”

Features

numeric features – defined for (nearly) every row

Clicks	Recency	Reading Level	Photo	Title
10	1.3	11	1	“New Tax Guidelines”
1000	1.7	3	1	“This 600lb baby…”
1000000	2.4	2	1	“18 reasons you should *never* look at this cat unless you…”
1	5.9	19	0	“The Brothers Karamazov: a neo-post-globalist perspective”

Features

boolean features – 0 or 1 ("dummy" variables)

Clicks	Recency	Reading Level	Photo	Title
10	1.3	11	1	"New Tax Guidelines"
1000	1.7	3	1	"This 600lb baby…"
1000000	2.4	2	1	"18 reasons you should *never* look at this cat unless you…"
1	5.9	19	0	"The Brothers Karamazov: a neo-post-globalist perspective"

Features

strings = boolean features — 0 or 1 ("dummy" variables)

Clicks	Recency	Reading Level	Photo	Title
10	1.3	11	1	"New Tax Guidelines"
1000	1.7	3	1	"This 600lb baby…"
1000000	2.4	2	1	"18 reasons you should *never* look at this cat unless you…"
1	5.9	19	0	"The Brothers Karamazov: a neo-post-globalist perspective"

Features

strings = boolean features – 0 or 1 ("dummy" variables)

Clicks	Recency	Reading Level	Photo	Title: "new"	Title: "tax"	Title: "this"	Title: "…"	…
10	1.3	11	1	1	0	0	0	…
1000	1.7	3	1	0	0	1	1	…
1000000	2.4	2	1	0	0	1	1	…
1	5.9	19	0	0	0	0	0	…

Features

“sparse features” – 0 for most rows

Clicks	Recency	Reading Level	Photo	Title: “new”	Title: “tax”	Title: “this”	Title: “…”	…
10	1.3	11	1	1	0	0	0	…
1000	1.7	3	1	0	0	1	1	…
1000000	2.4	2	1	0	0	1	1	…
1	5.9	19	0	0	0	0	0	…

Clicker Question!

Clicker Question!

For the problem set up, how many features will there be? I.e. how many columns in our X matrix, (not including Y)?

```
Y: happiness
X1: day of week (“monday”, “tuesday”, … “sunday”)
X2: bank account balance (real value)
X3: breakfast (yes,no)
X4: whether you have found your inner peace
(yes,no)
X5: words from last week’s worth of tweets
(assuming tweets are at most 15 words long and
there are 100K words in the English vocabulary)
```

(a) 112,000

(b) 5

(c) 27

(d) 110,000

Clicker Question!

For the problem set up, how many features will there be? I.e. how many columns in our X matrix, (not including Y)?

```
Y: happiness
X1: day of week ("monday", "tuesday", … "sunday") 7
X2: bank account balance (real value) 1
X3: breakfast (yes,no) 1
X4: whether you have found your inner peace 1
(yes,no)
X5: words from last week's worth of tweets  100,000
(assuming tweets are at most 15 words long and
there are 100K words in the English vocabulary)
```

(a) 100,012
(b) 5
(c) 27
(d) 100,010

Defining an ML problem

Objective/Loss Function = squared difference between predicted total number of clicks and actual total number of clicks

~~Task = Increase Consumption~~

Model

~~Data = Reading Habits~~

Features = ???

Defining an ML problem

Objective/Loss Function = squared difference between predicted total number of clicks and actual total number of clicks

~~Task = Increase Consumption~~

↑

Model

↑

~~Data = Reading Habits~~

Features = {Recency:float, ReadingLevel:Int, Photo:Bool, Title_New:Bool, Title_Tax:Bool, ...}

Defining an ML problem

Objective/Loss Function = squared difference between predicted total number of clicks and actual total number of clicks

~~Task = Increase Consumption~~

Model

~~Data = Reading Habits~~

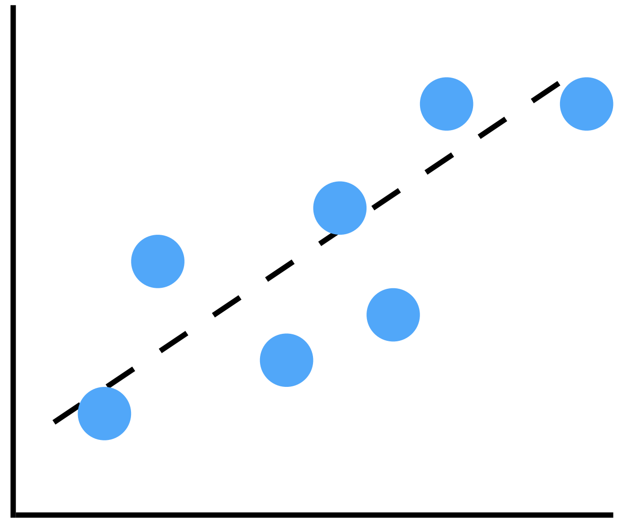
Features = {Recency:float, ReadingLevel:Int, Photo:Bool, Title_New:Bool, Title_Tax:Bool, ...}

Model

ML = Function Approximation

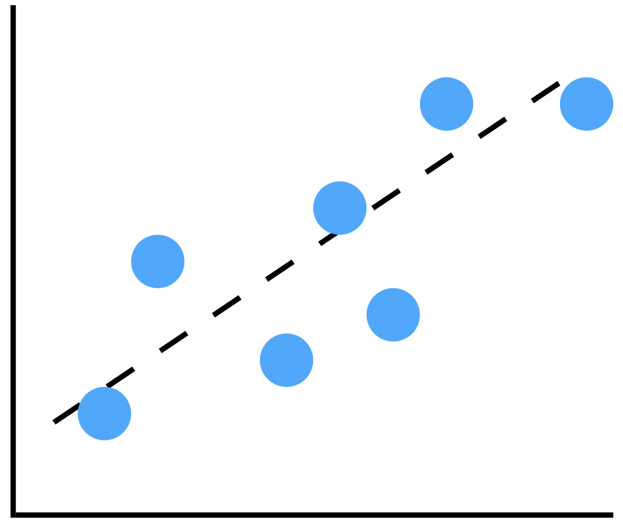
Model

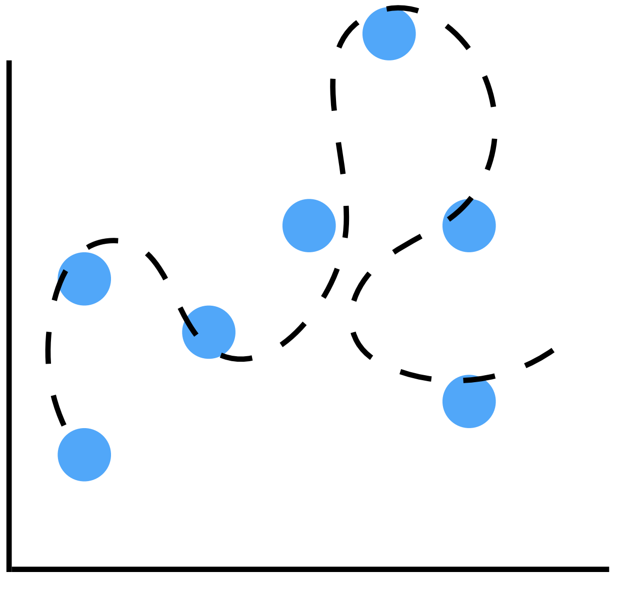
ML = Function Approximation



Model

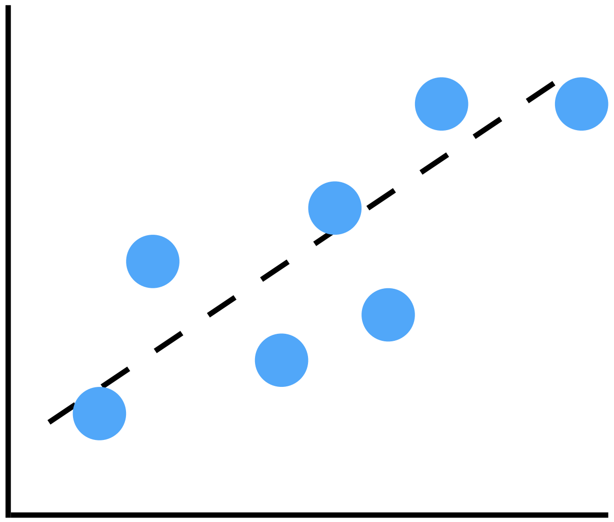
ML = Function Approximation

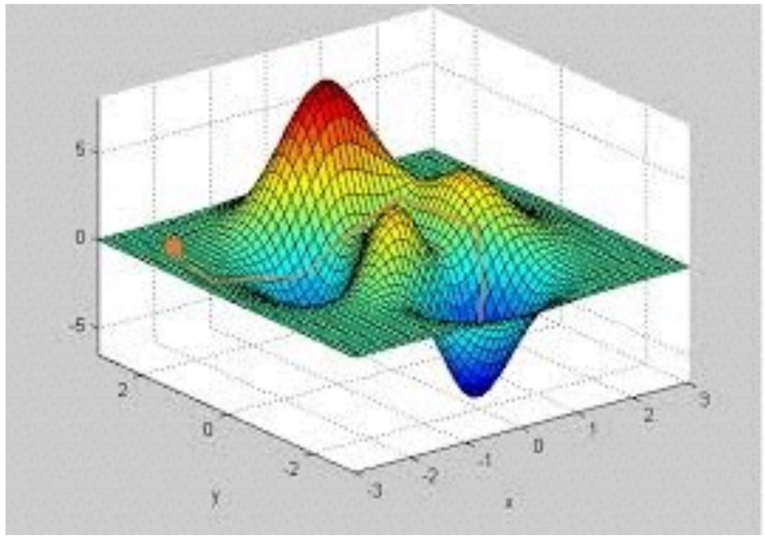


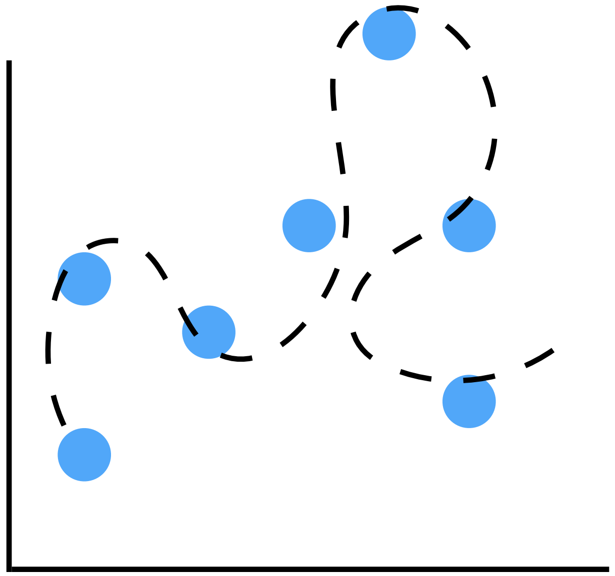


Model

ML = Function Approximation

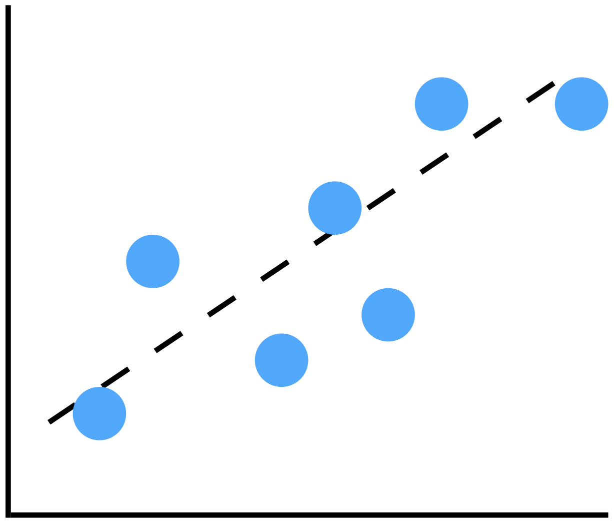


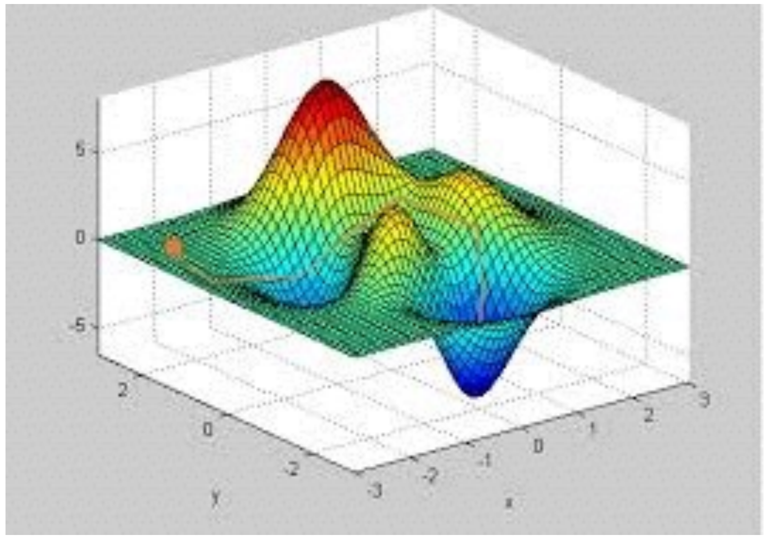


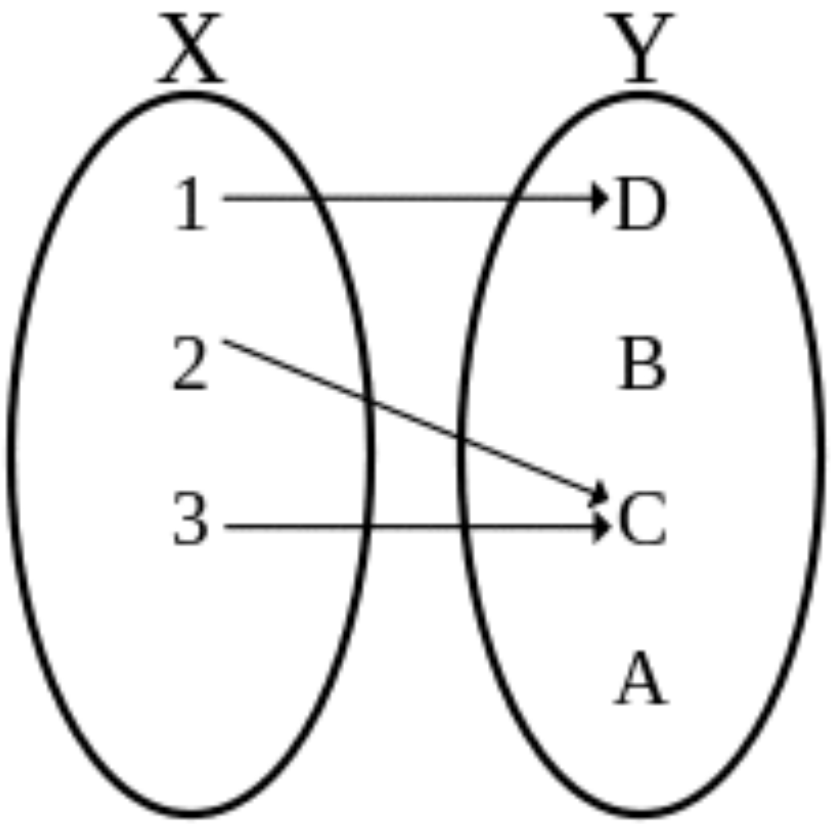


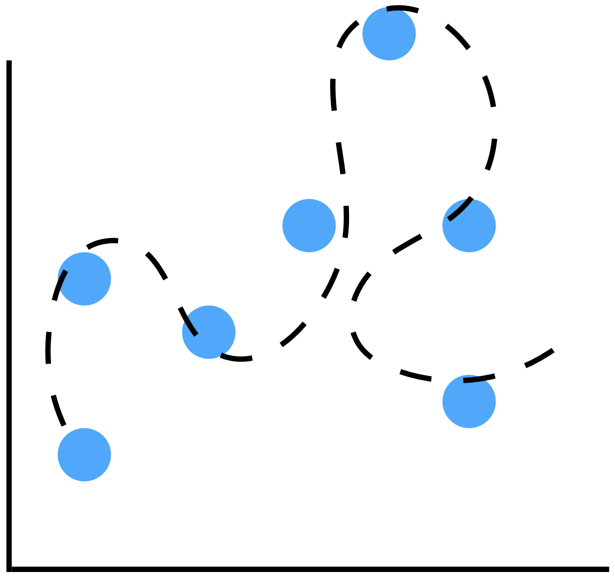
Model

ML = Function Approximation



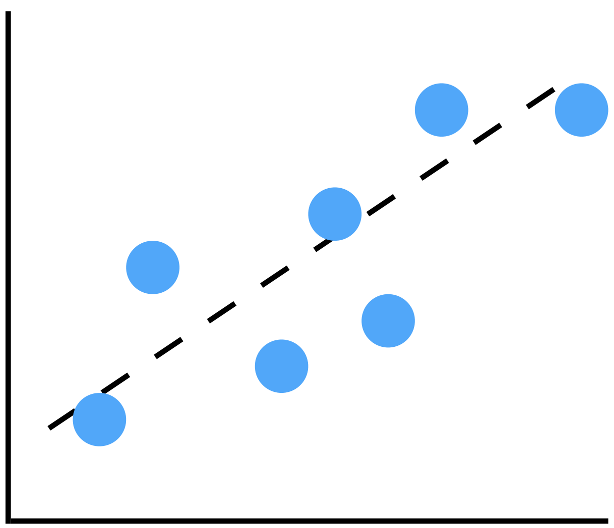


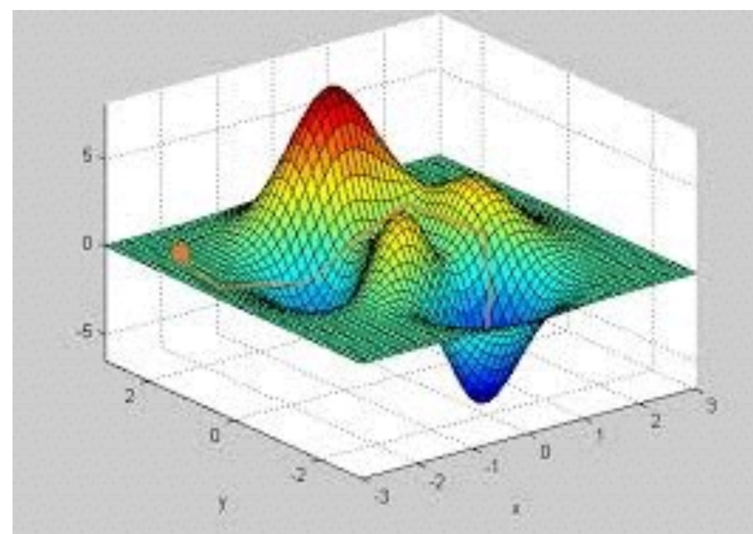




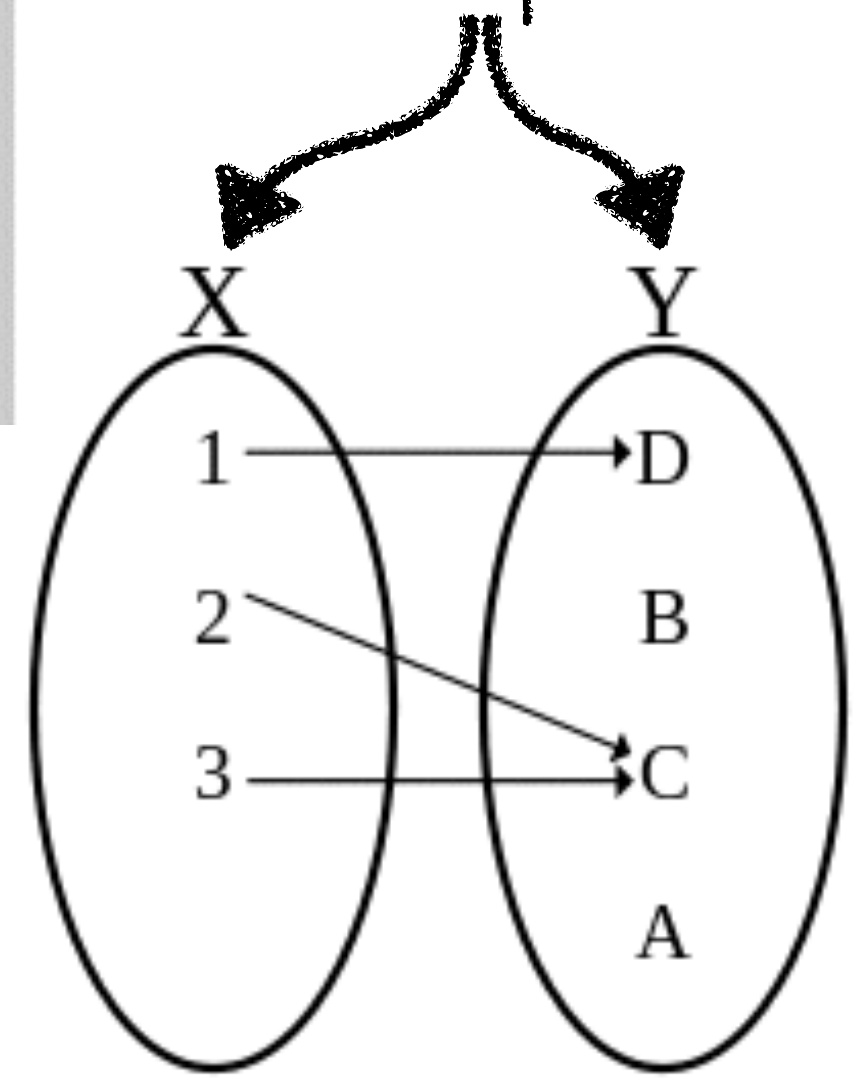
Model

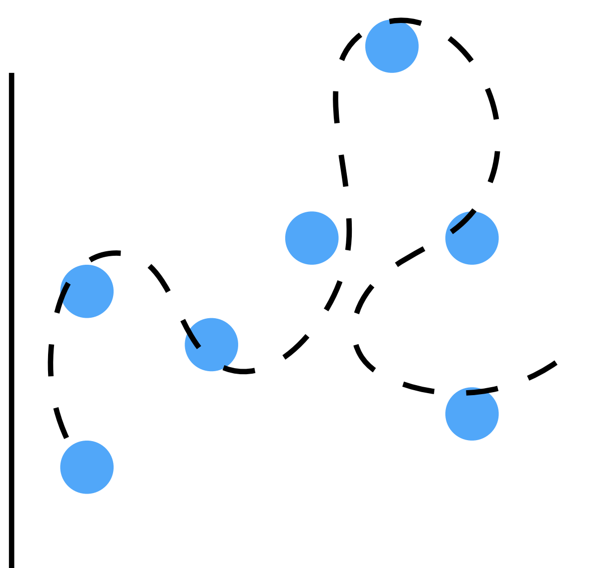
ML = Function Approximation





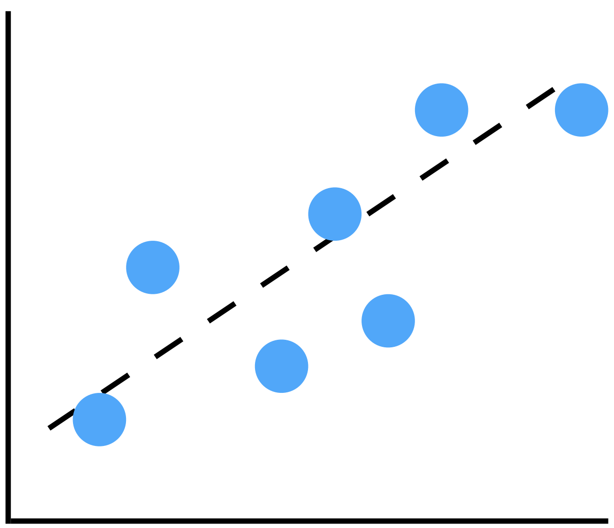
You define inputs and outputs.

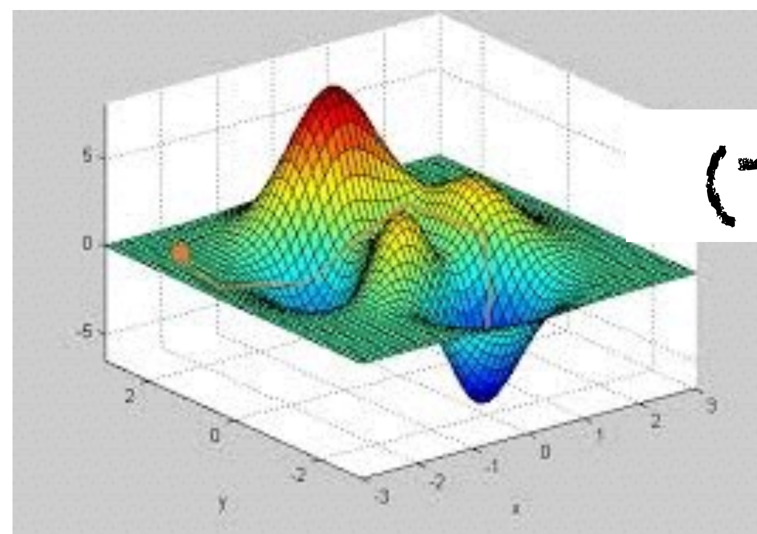




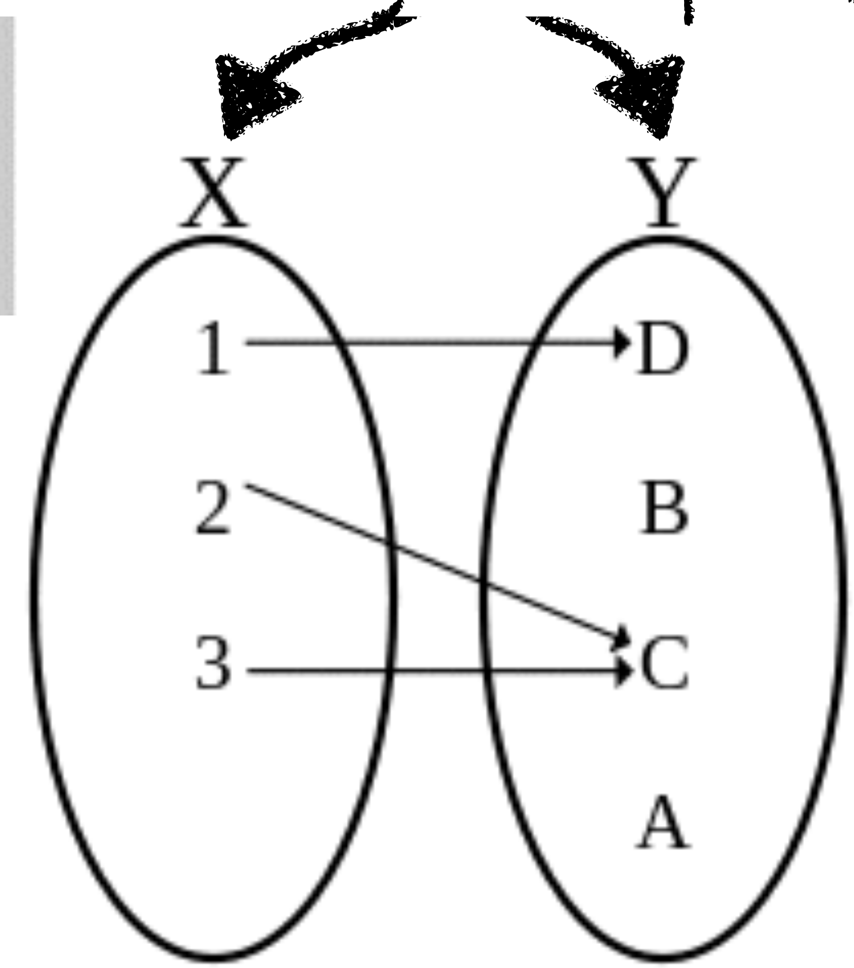
Model

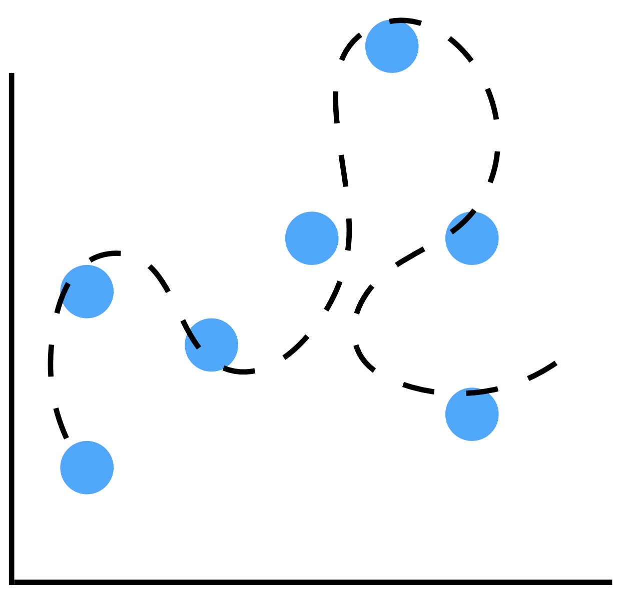
ML = Function Approximation





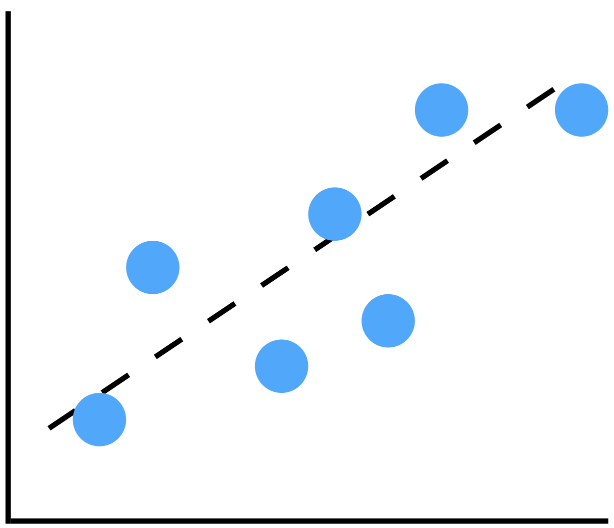
You define inputs
and outputs.
(The really hard part)

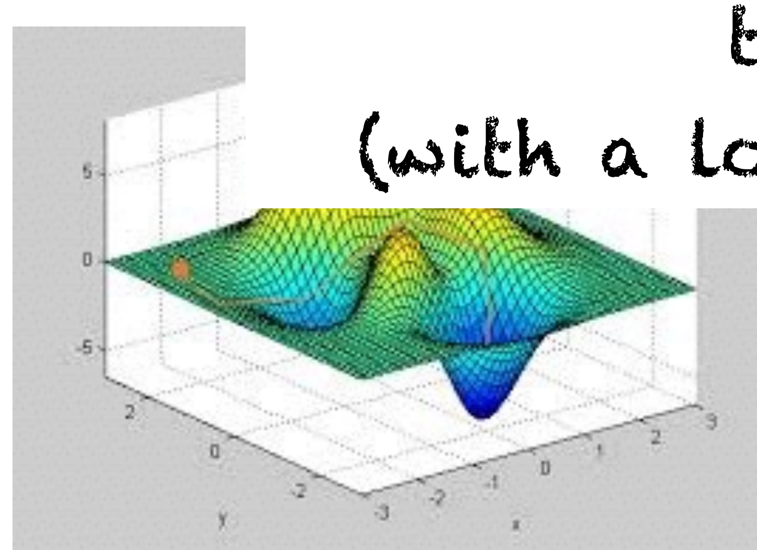




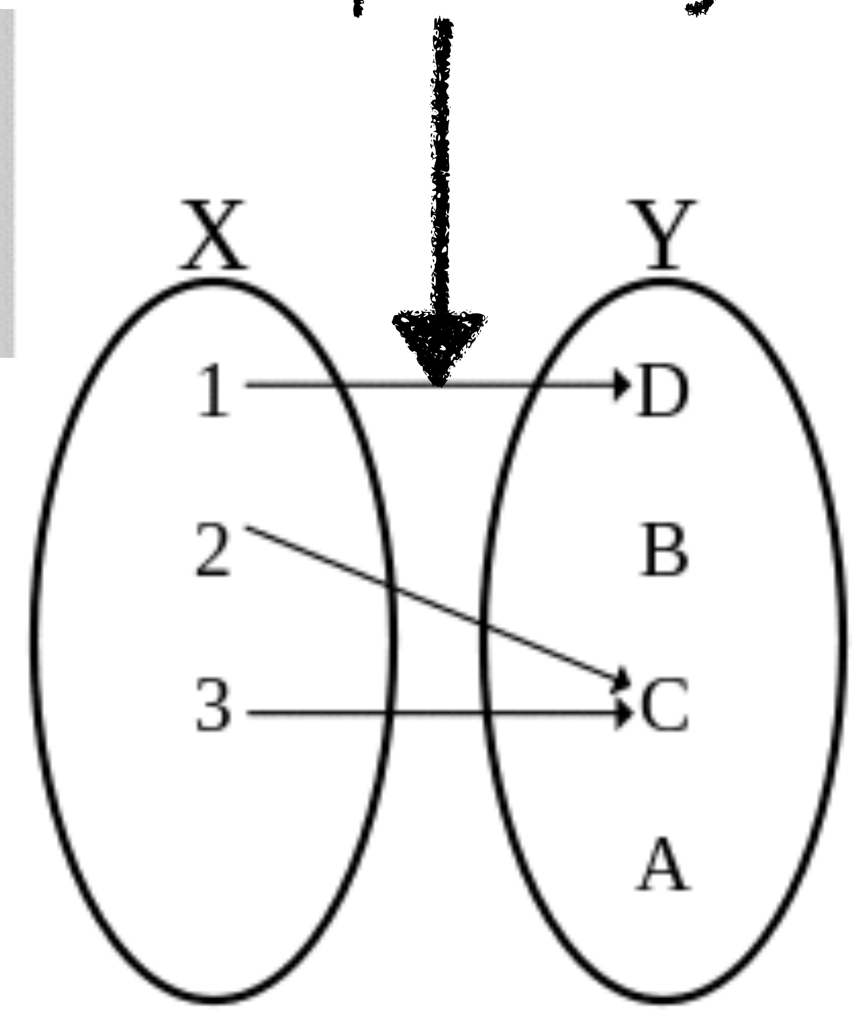
Model

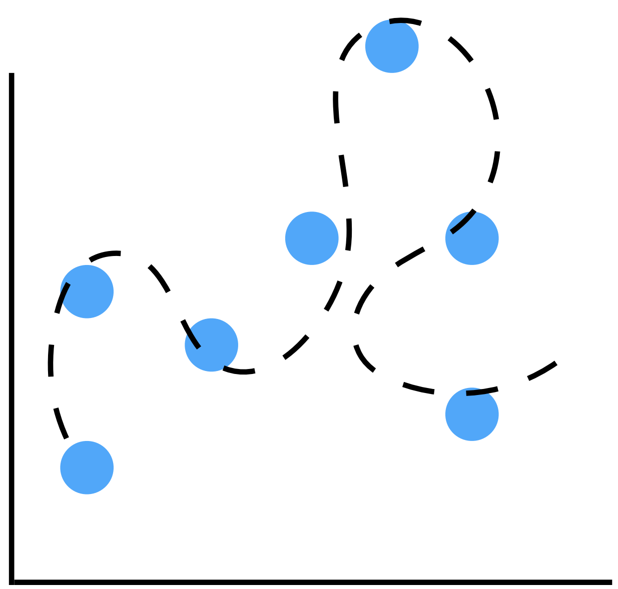
ML = Functic





The machine will (ideally) learn
the function
(with a lot of help from you)

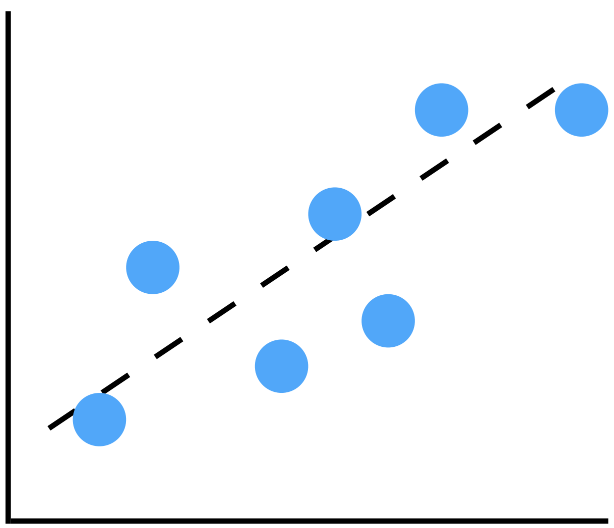


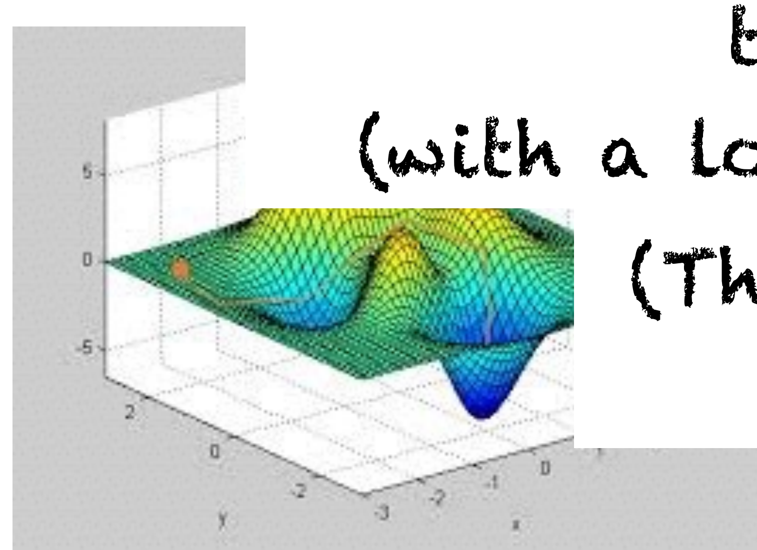


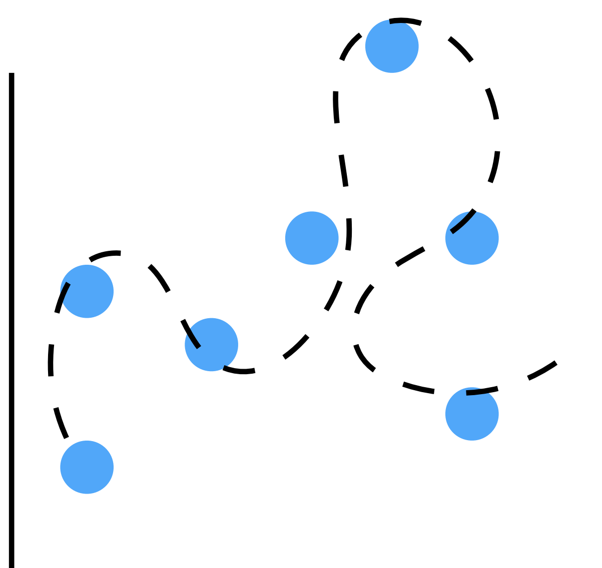
Model

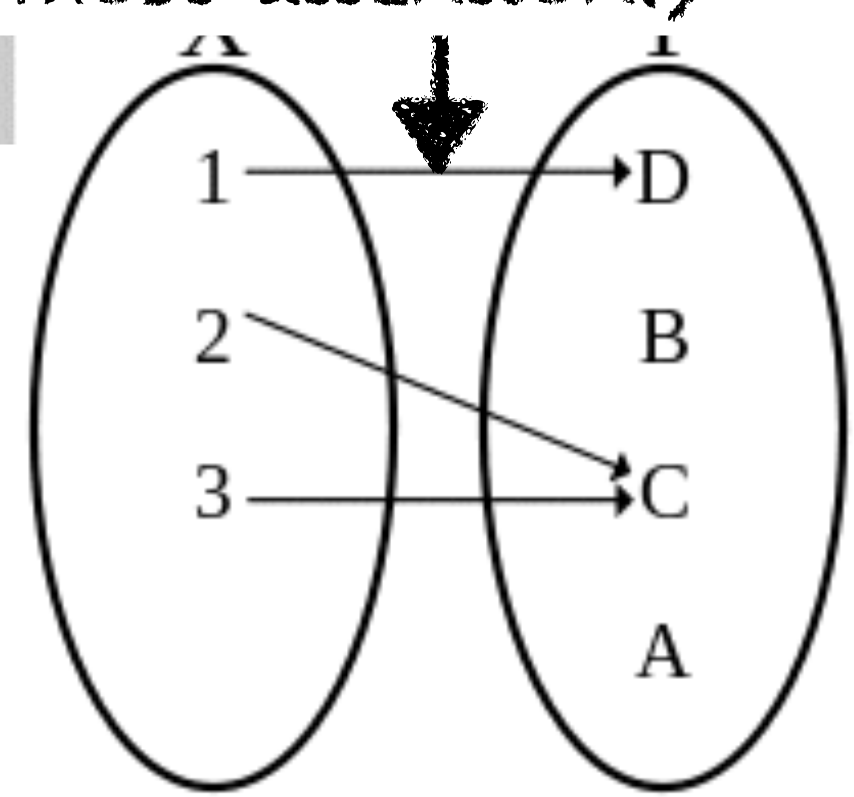
ML = Functic

The machine will (ideally) learn
the function
(with a lot of help from you)
(The part that gets the
most attention.)









Model

#1

- Make assumptions about the problem domain.

Model

#1

- Make assumptions about the problem domain.
- How is the data generated?

Model

#1

- Make assumptions about the problem domain.
- How is the data generated?
- How is the decision-making procedure structured?

Model

#1

- Make assumptions about the problem domain.
- How is the data generated?
- How is the decision-making procedure structured?
- What types of dependencies exist?

Model

#1

- Make assumptions about the problem domain.
- How is the data generated?
- How is the decision-making procedure structured?
- What types of dependencies exist?
- Trending buzzword: “inductive biases”

Model

#1
- Make assumptions about the problem domain.
- How is the data generated?
- How is the decision-making procedure structured?
- What types of dependencies exist?
- Trending buzzword: “inductive biases”

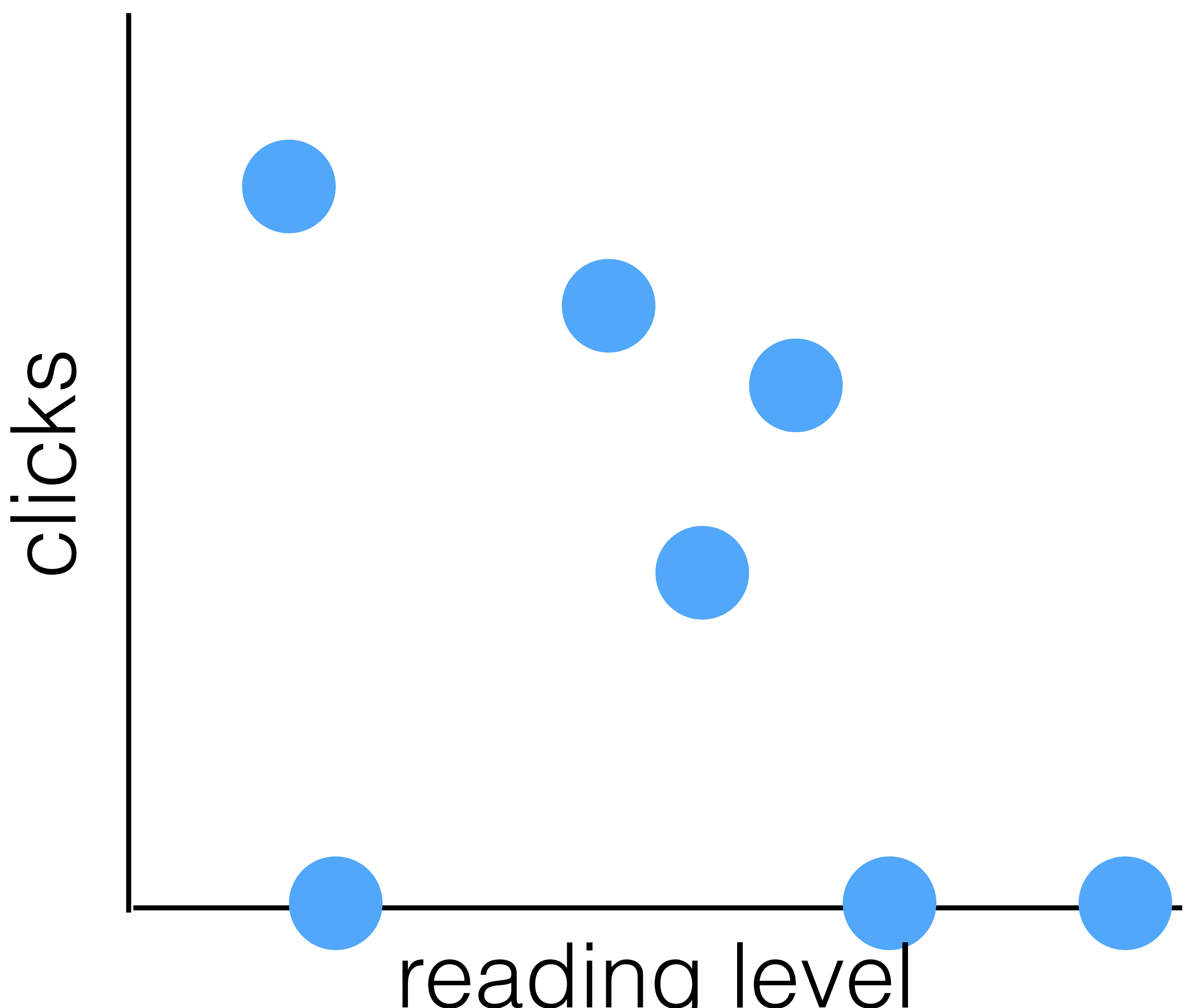
#2
- How to train the model?

Model

#1
- Make assumptions about the problem domain.
- How is the data generated?
- How is the decision-making procedure structured?
- What types of dependencies exist?
- Trending buzzword: “inductive biases”

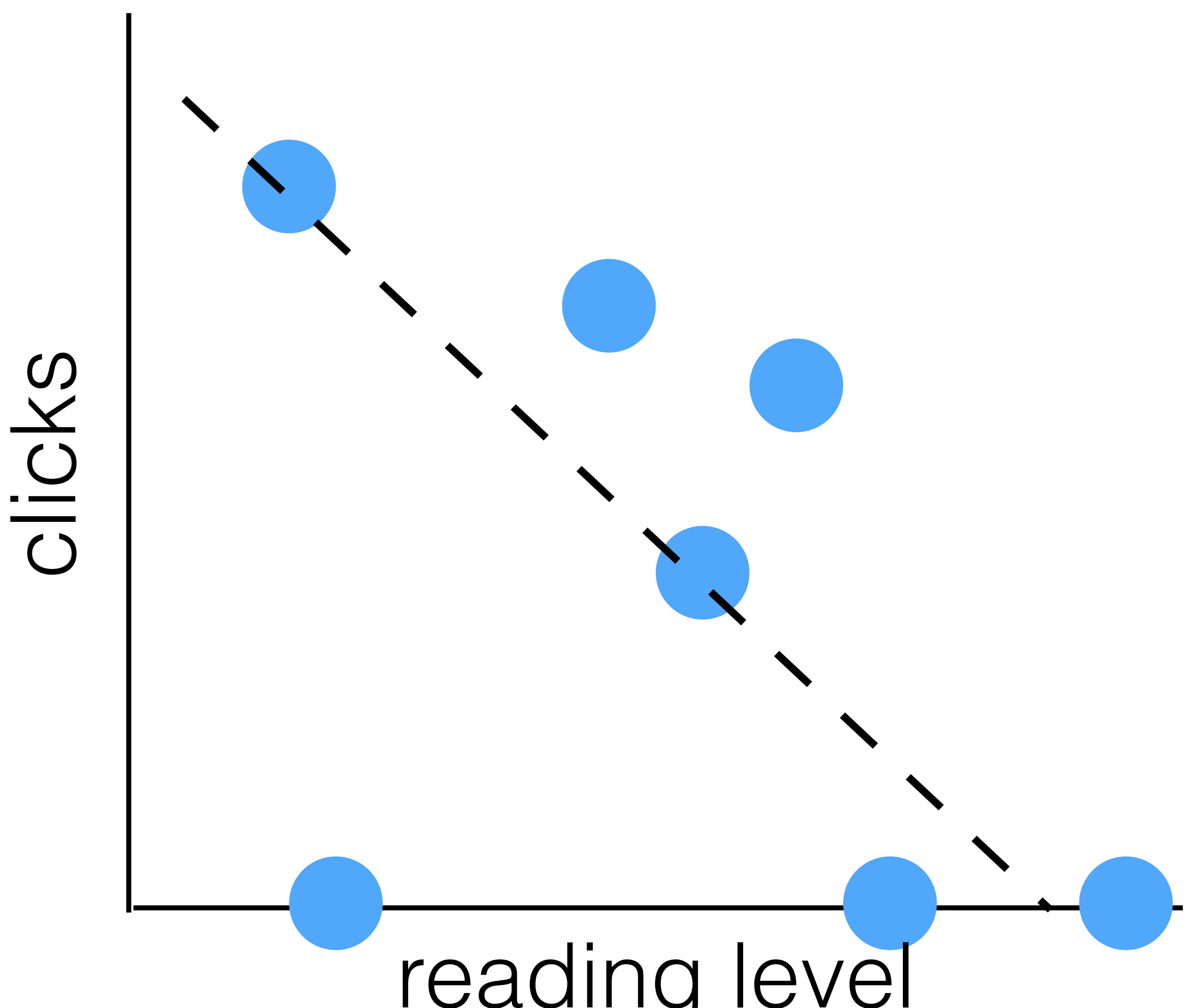
#2
- How to train the model?

Model



Model

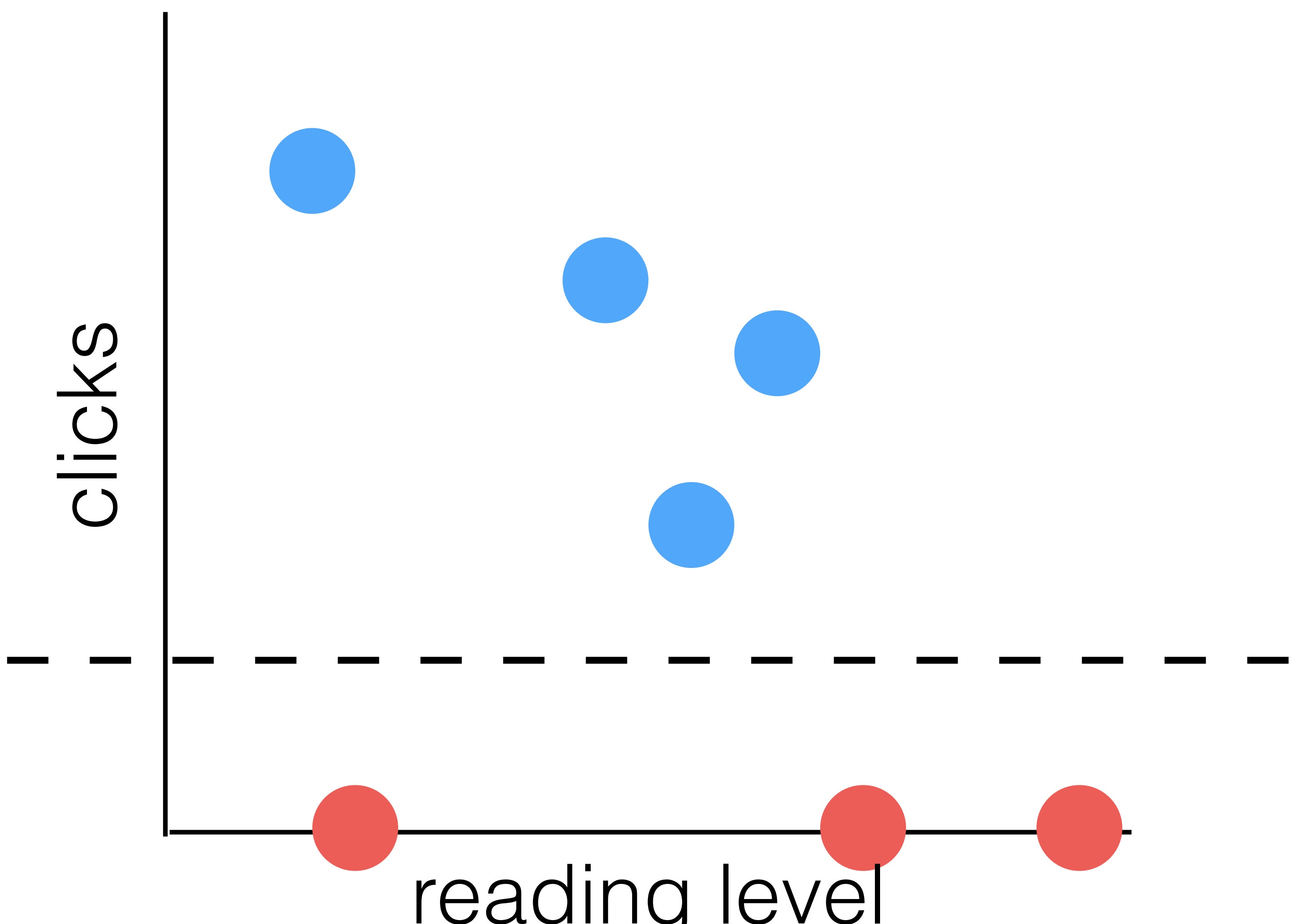
Regression: continuous (infinite) output
f(reading level) = # of clicks



Model

Classification: discrete (finite) output

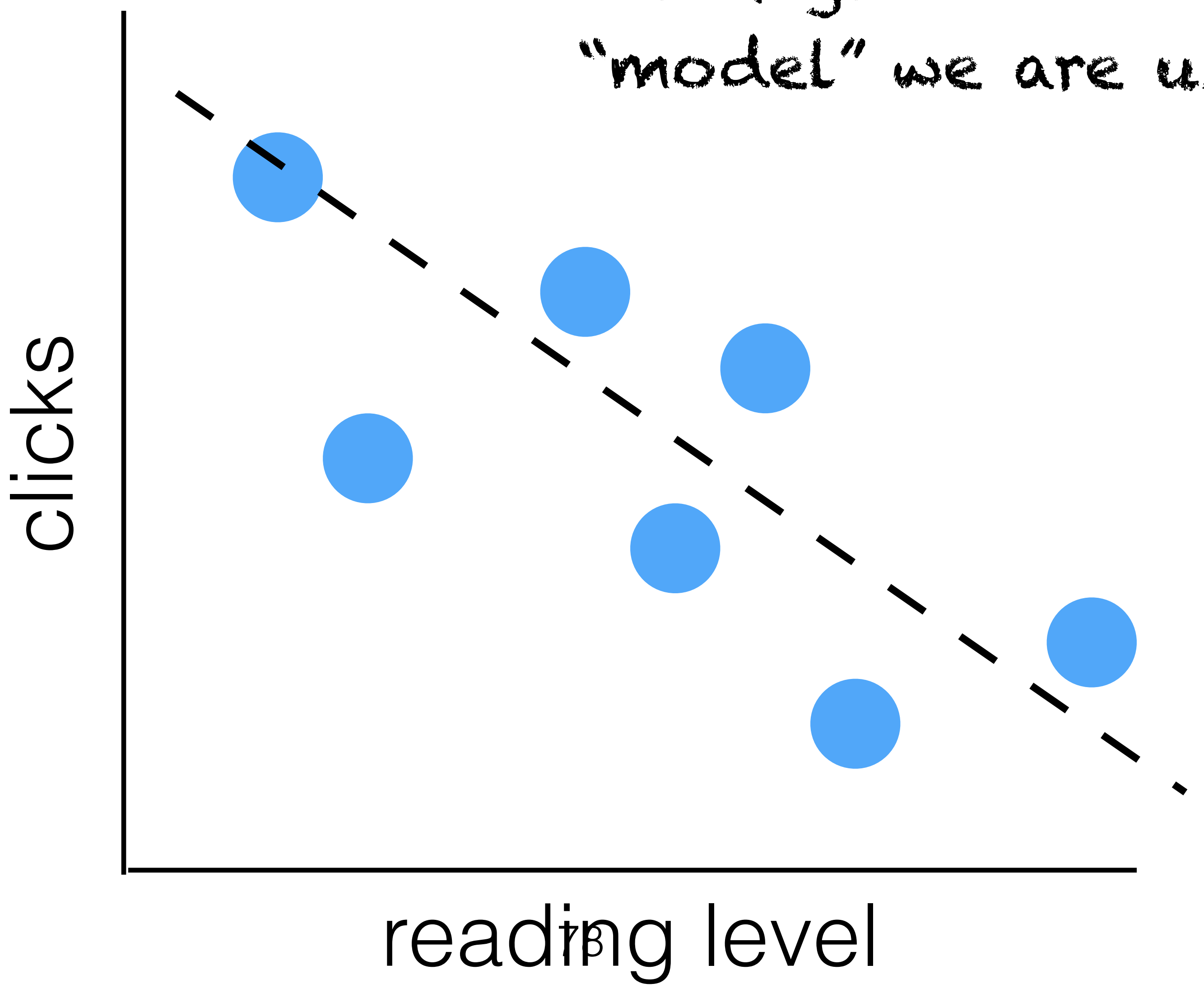
f(reading level) = {clicked, not clicked}



Model

$$clicks = m(reading\_level) + b$$

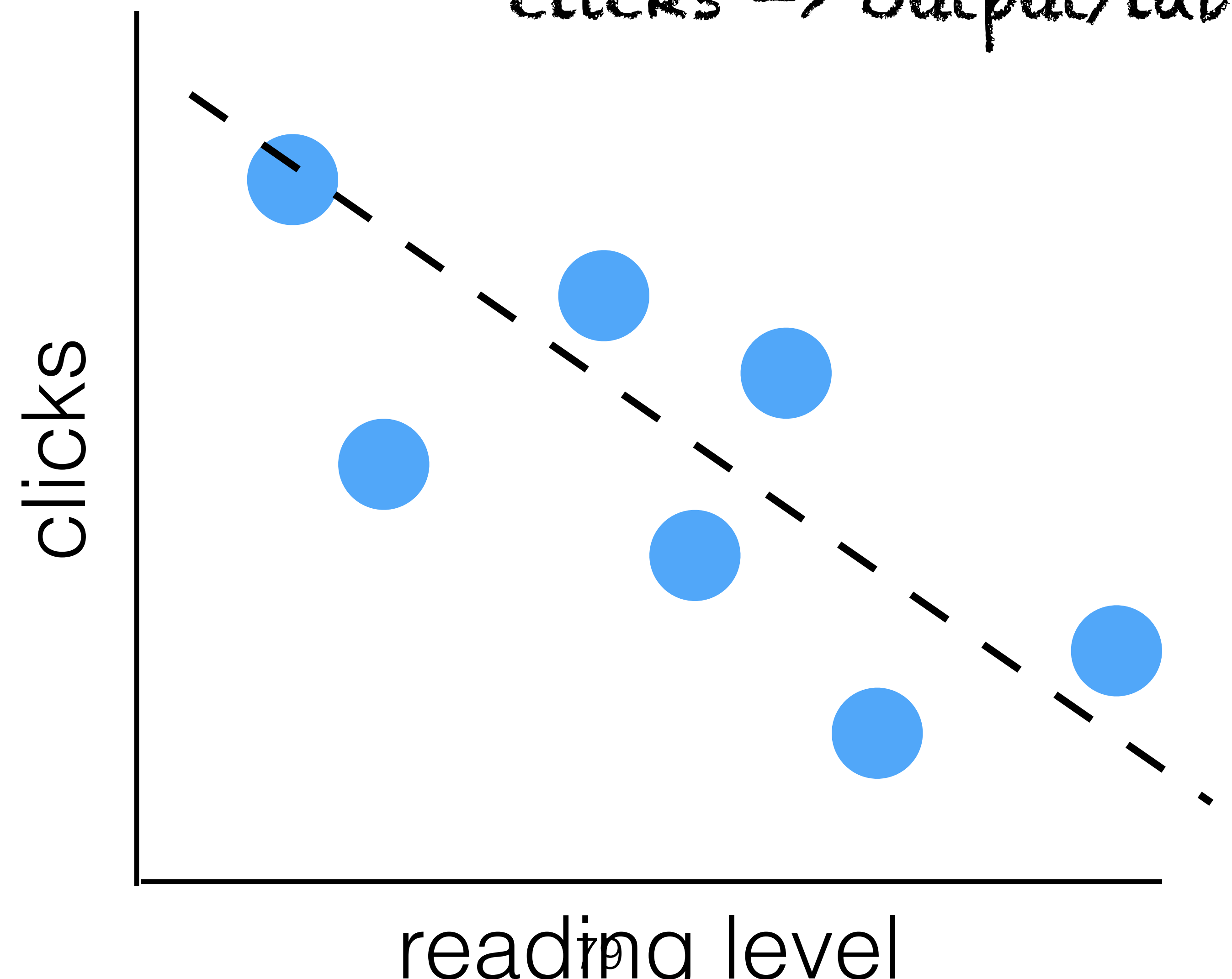
Linear Regression -> The specific "model" we are using here.



Model

`clicks = m(reading_level) + b`

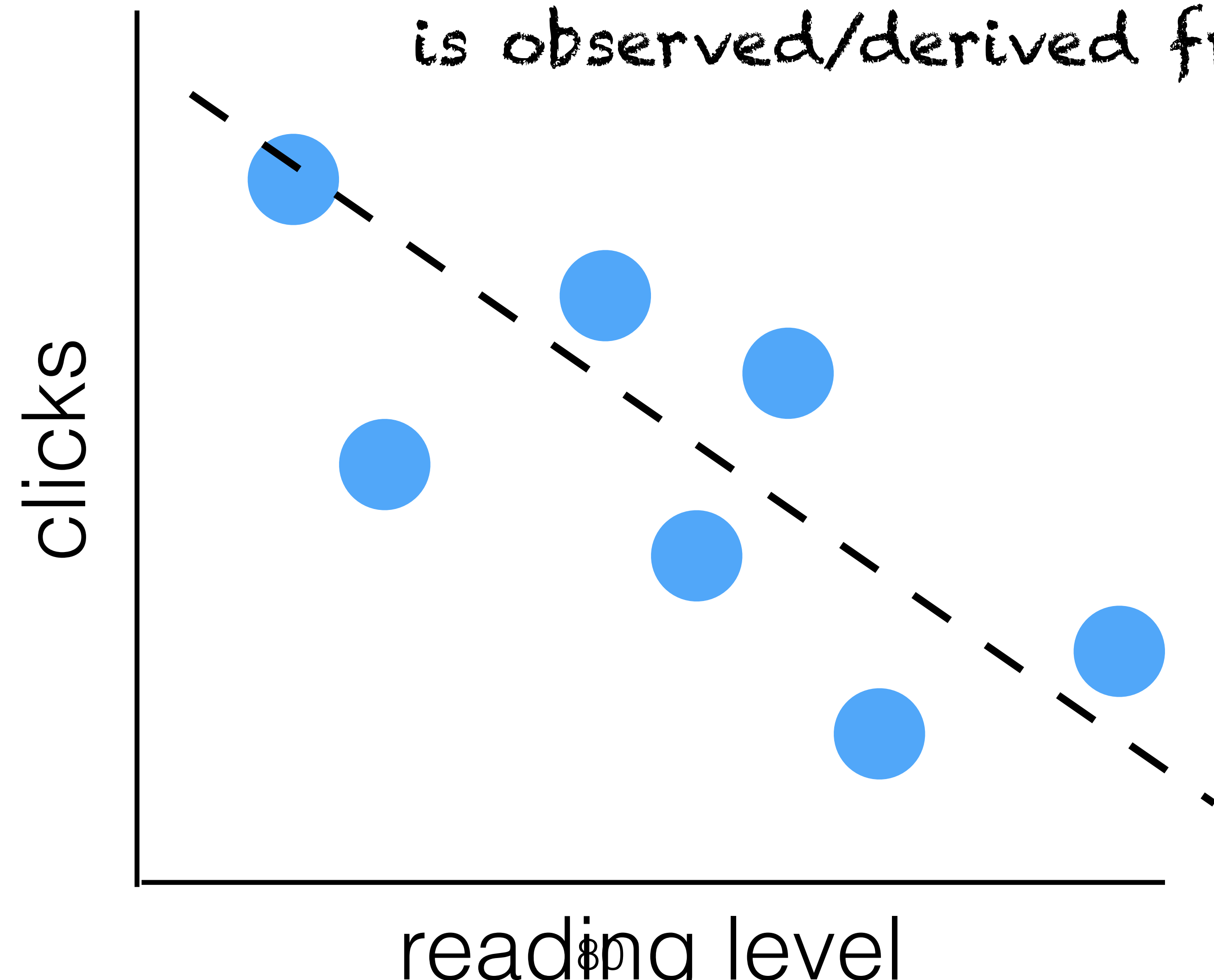
clicks → output/labels/target



Model

```
clicks = m(reading_level) + b
```

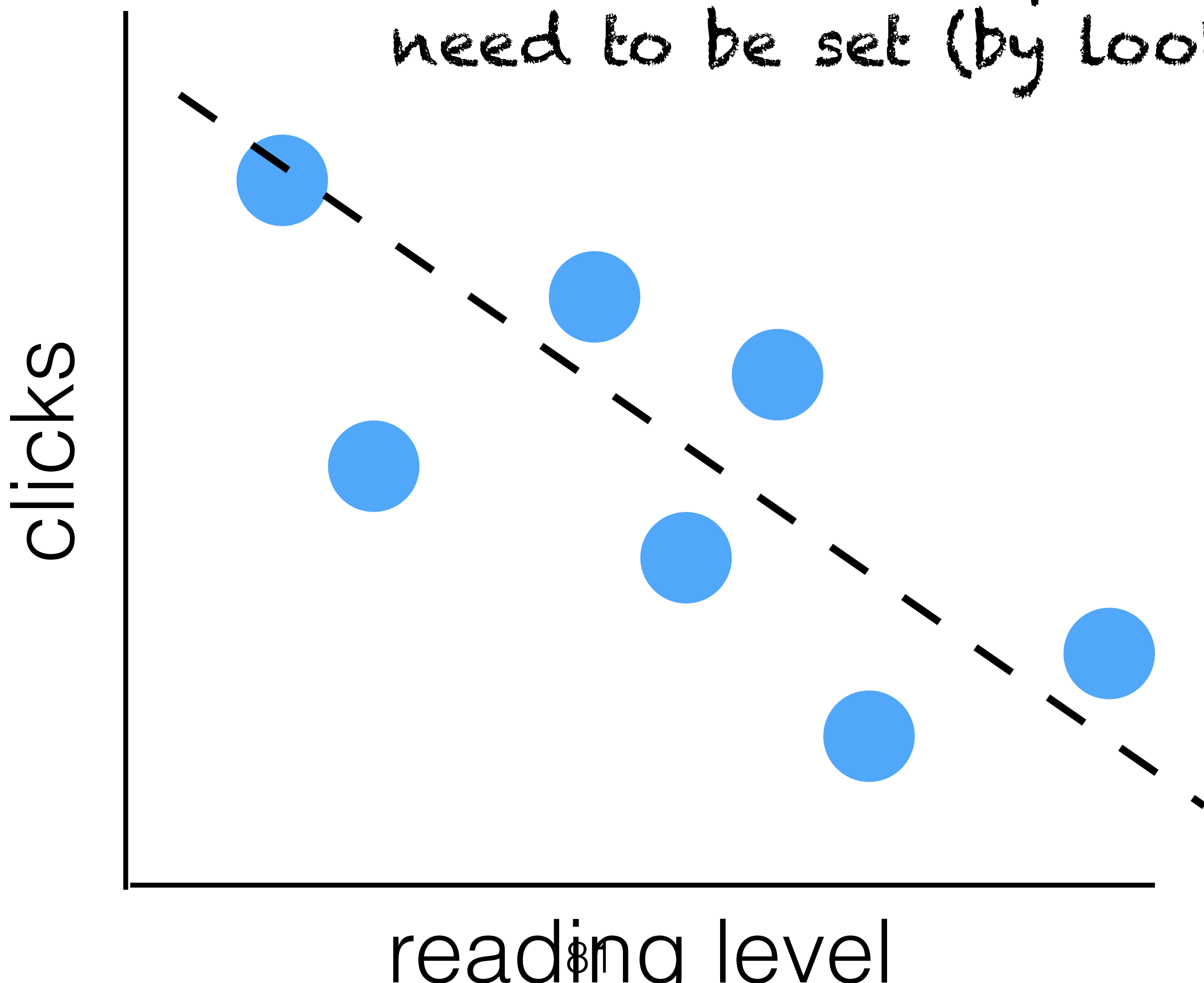
reading level → The "feature" which is observed/derived from the data



Model

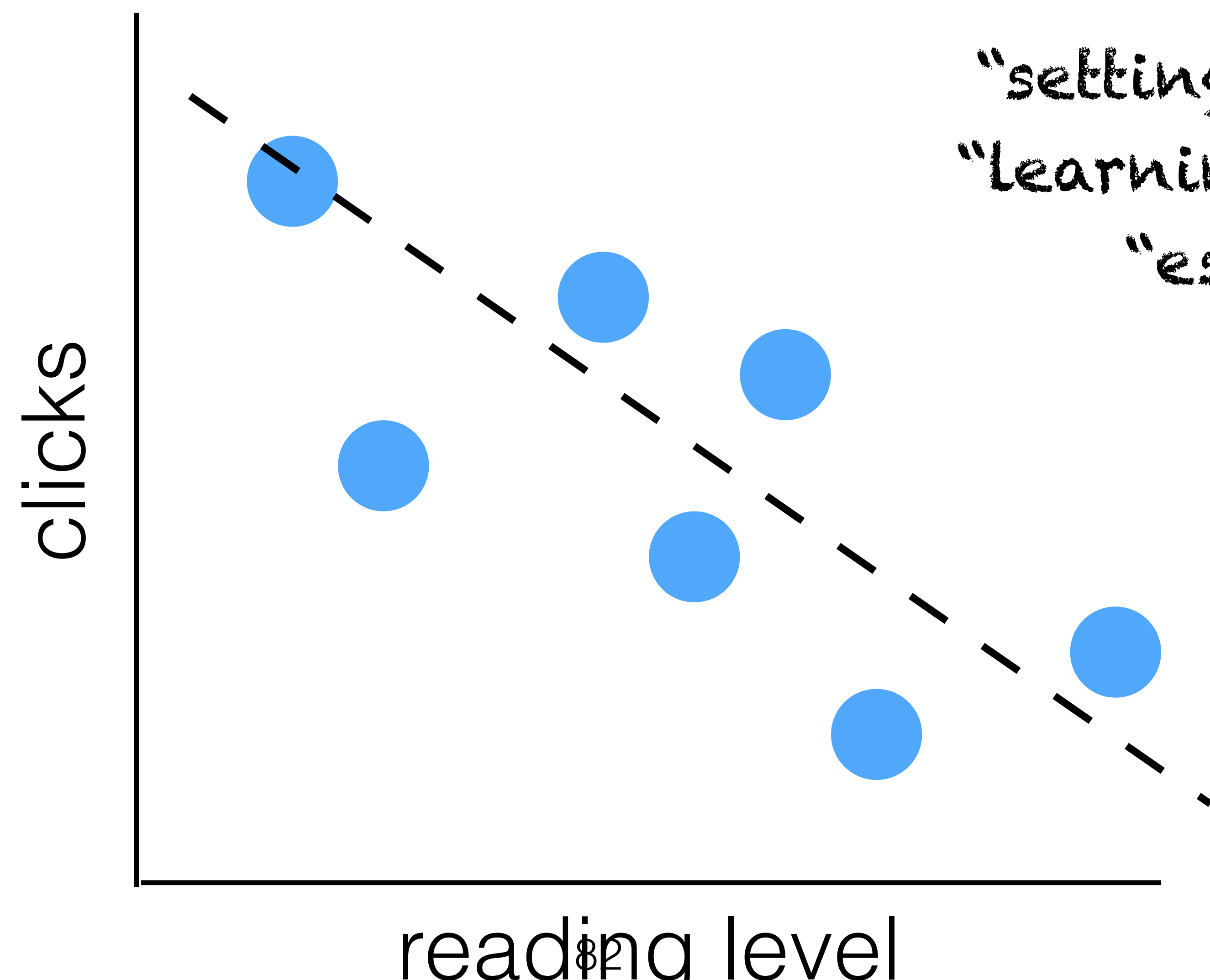
`clicks = m(reading_level) + b`

m and b -> The "parameters" which need to be set (by looking at data)



Model

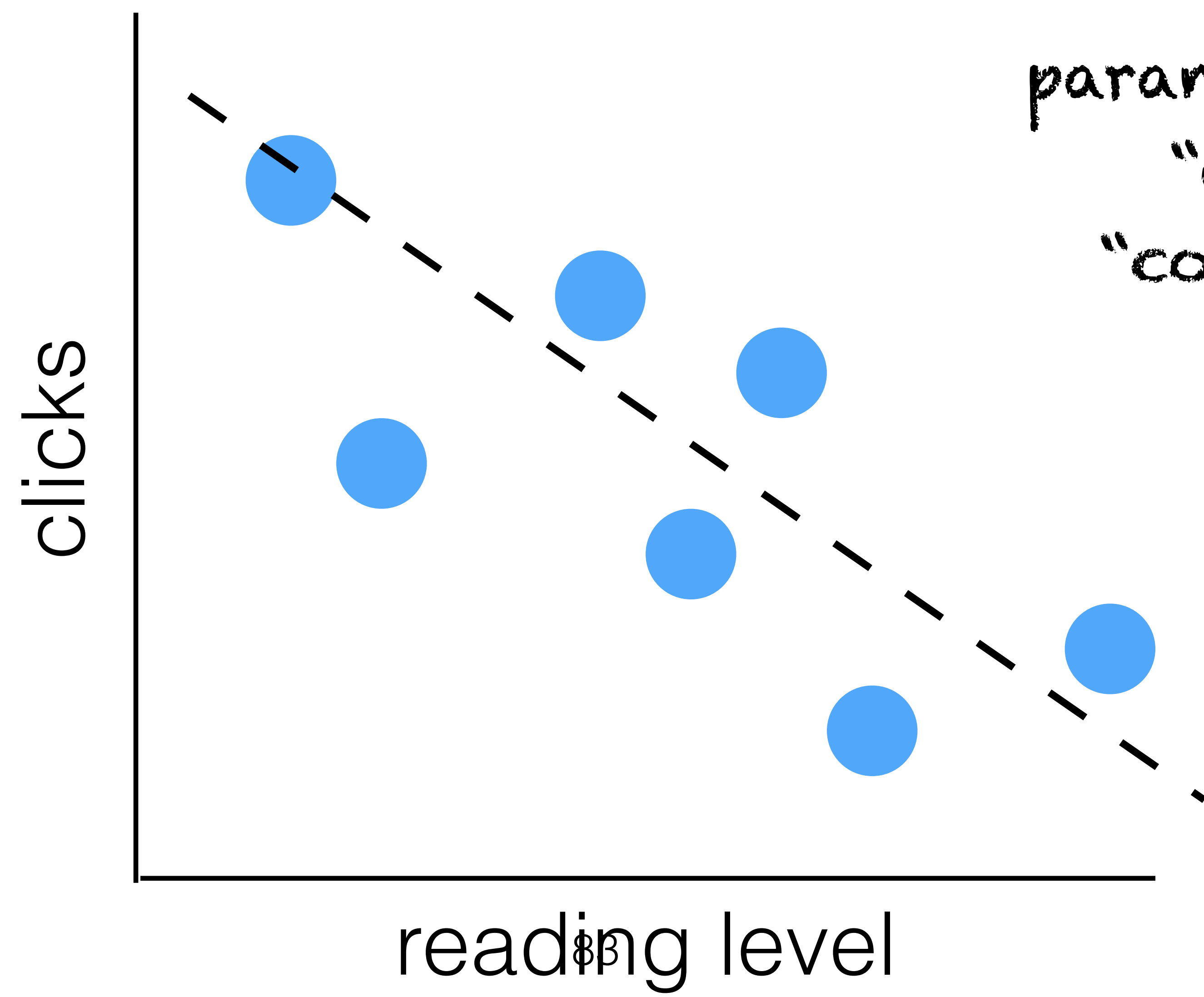
```
clicks = m(reading_level) + b
m = cov(rl, c)/var(rl)
```



"setting parameters", "learning", "training", "estimation"

Model

```
clicks = m(reading_level) + b
m = -2.4
```



parameter values,
"weights",
"coefficients"

Defining an ML problem

Objective/Loss Function = squared difference between predicted total number of clicks and actual total number of clicks

~~Task = Increase Consumption~~

~~Model~~

Linear Regression

~~Data = Reading Habits~~

Features = {Recency:float, ReadingLevel:Int, Photo:Bool, Title_New:Bool, Title_Tax:Bool, ...}

Defining an ML problem

Objective/Loss Function = squared difference between predicted total number of clicks and actual total number of clicks

↑

Linear Regression

↑

Features = {Recency:float, ReadingLevel:Int, Photo:Bool, Title_New:Bool, Title_Tax:Bool, …}

Defining an ML problem

Objective/Loss Function = squared difference between predicted total number of clicks and actual total number of clicks

Linear Regression

Features = {Recency:float, Reac
Photo:Bool, Title_New:Bool, Title_

Defining an ML problem

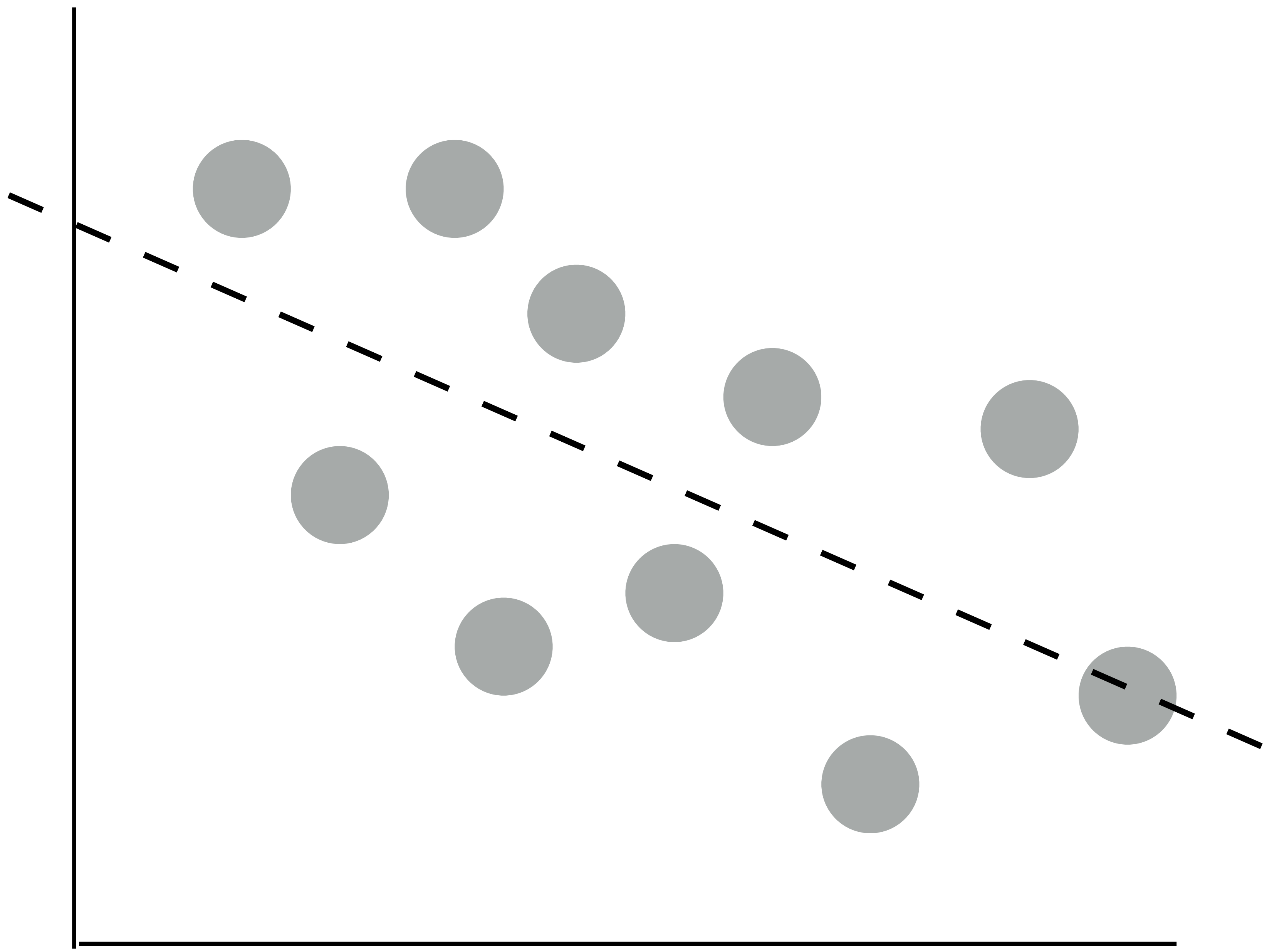
Objective/Loss Function = squared difference between predicted total number of clicks and actual total number of clicks

Soooo...how do I know if my model is good?

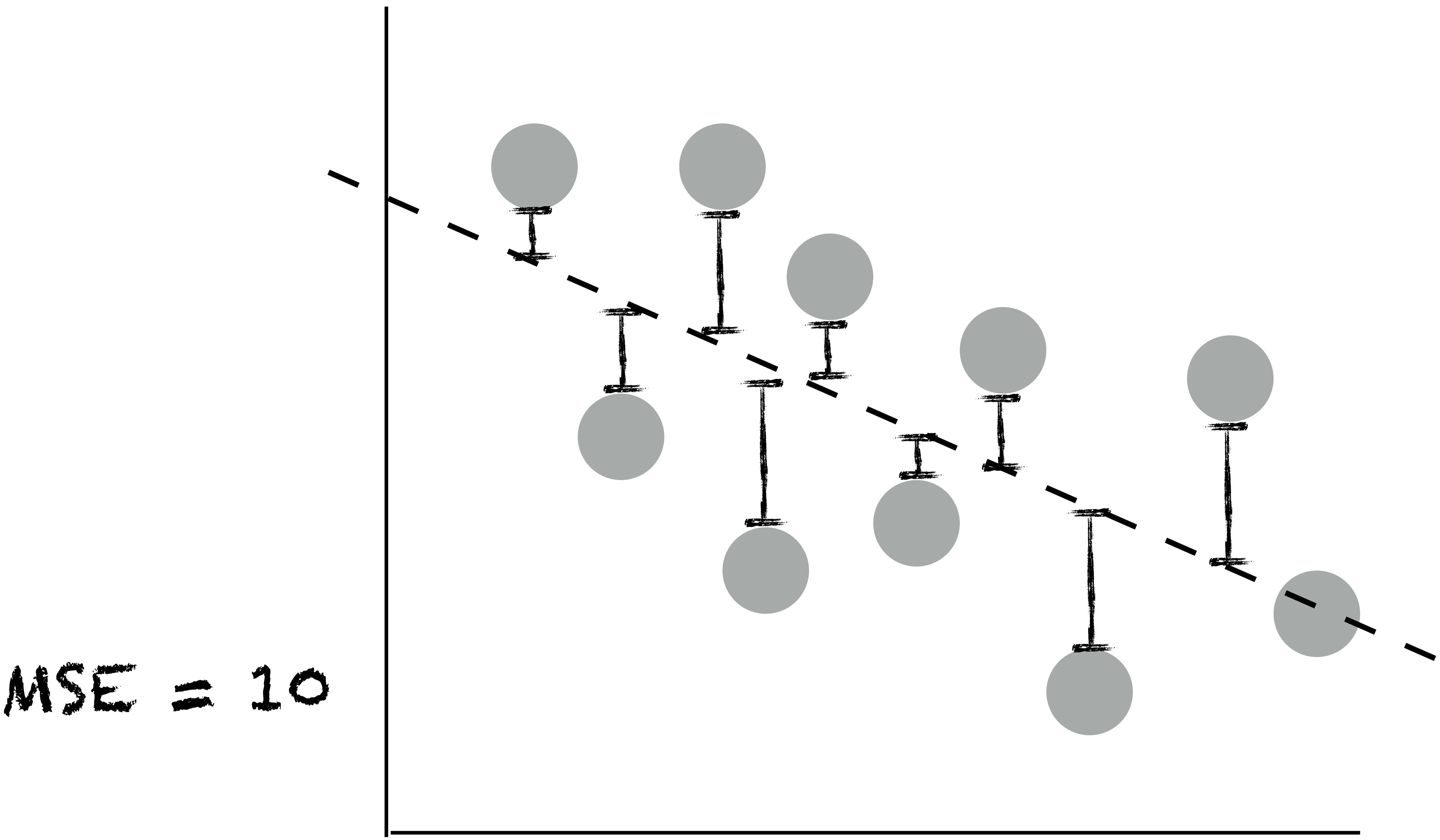
Linear Regression

Features = {Recency:float, Reac
Photo:Bool, Title_New:Bool, Title_

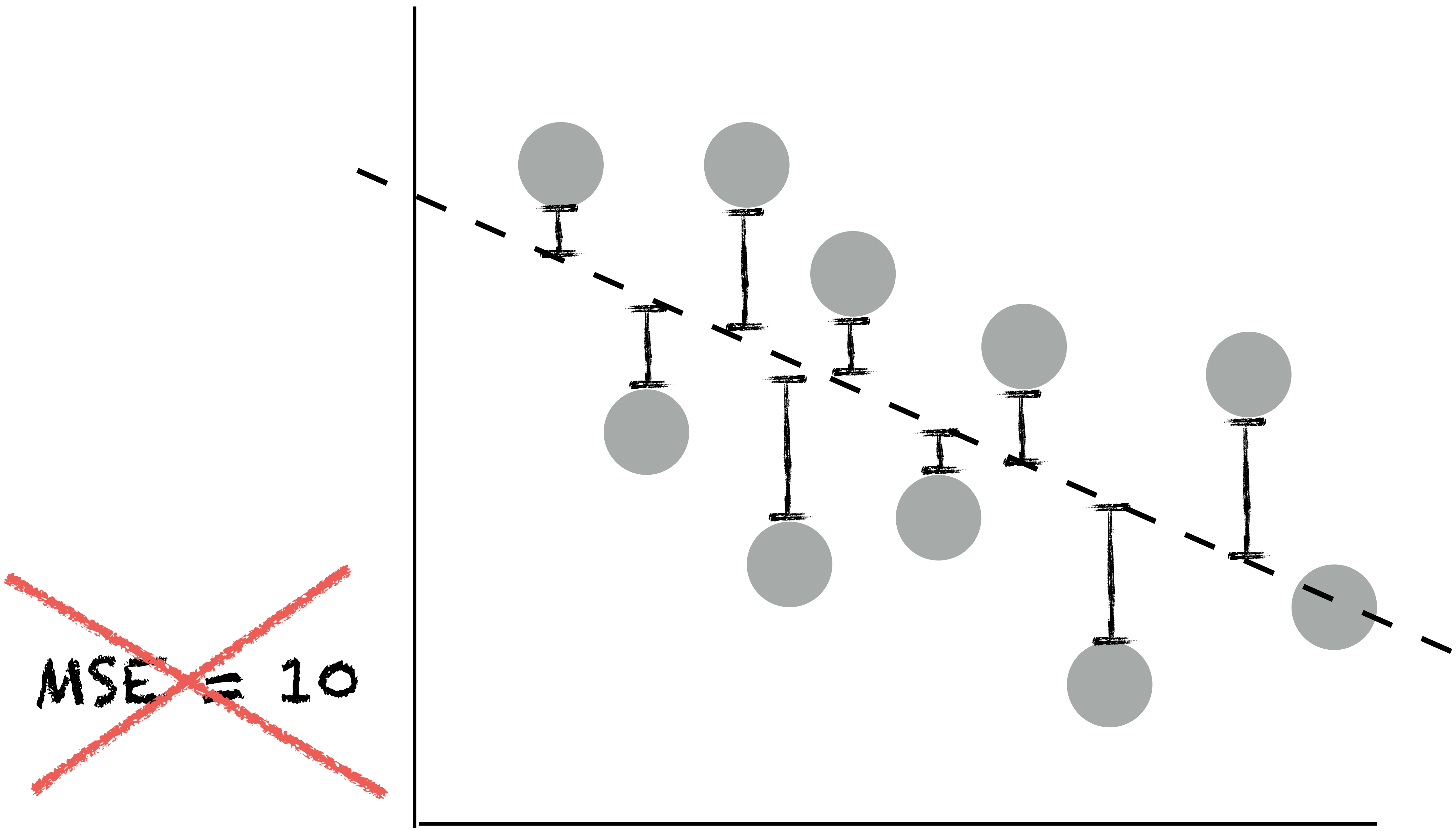
Train/Test Splits



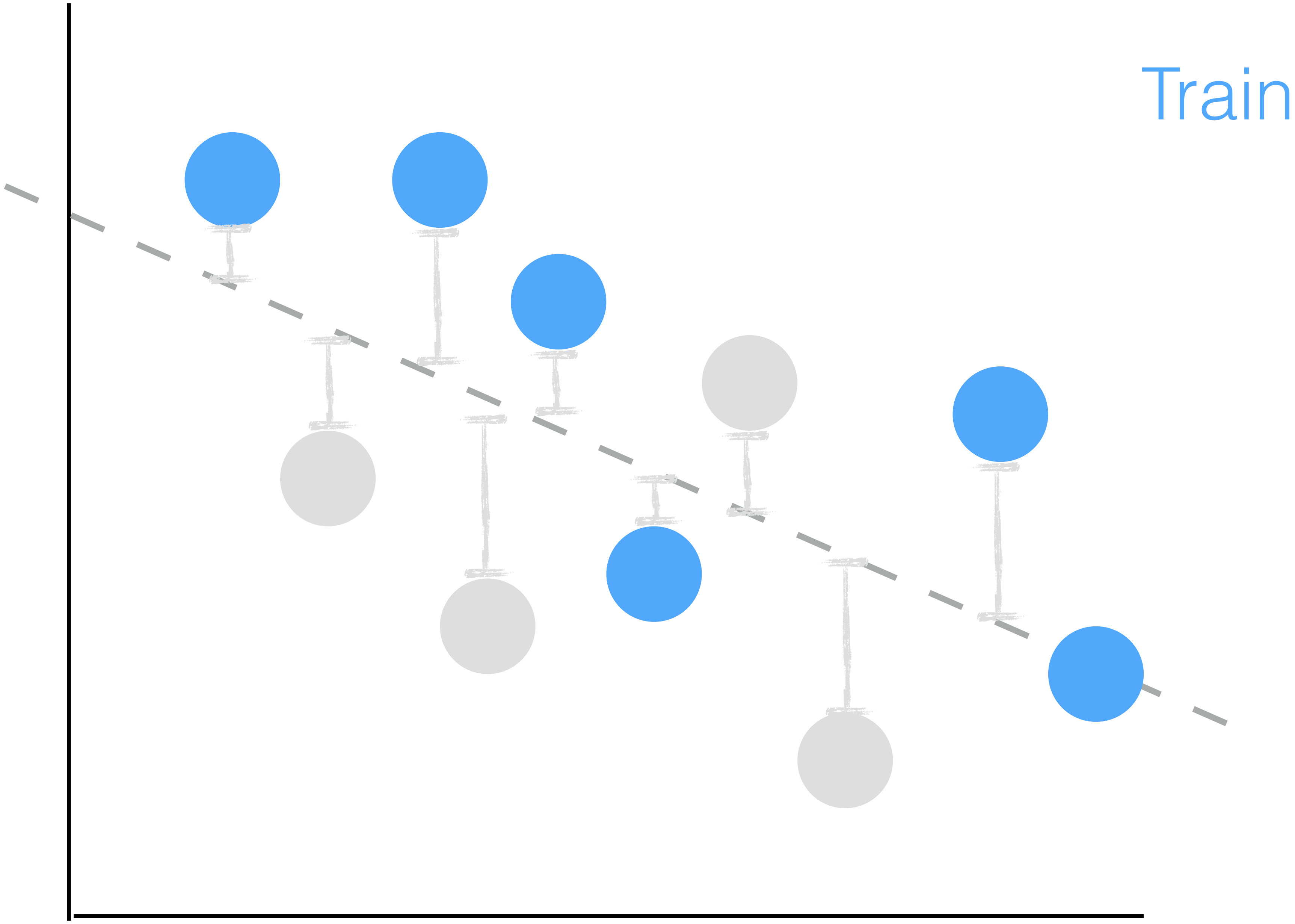
Train/Test Splits



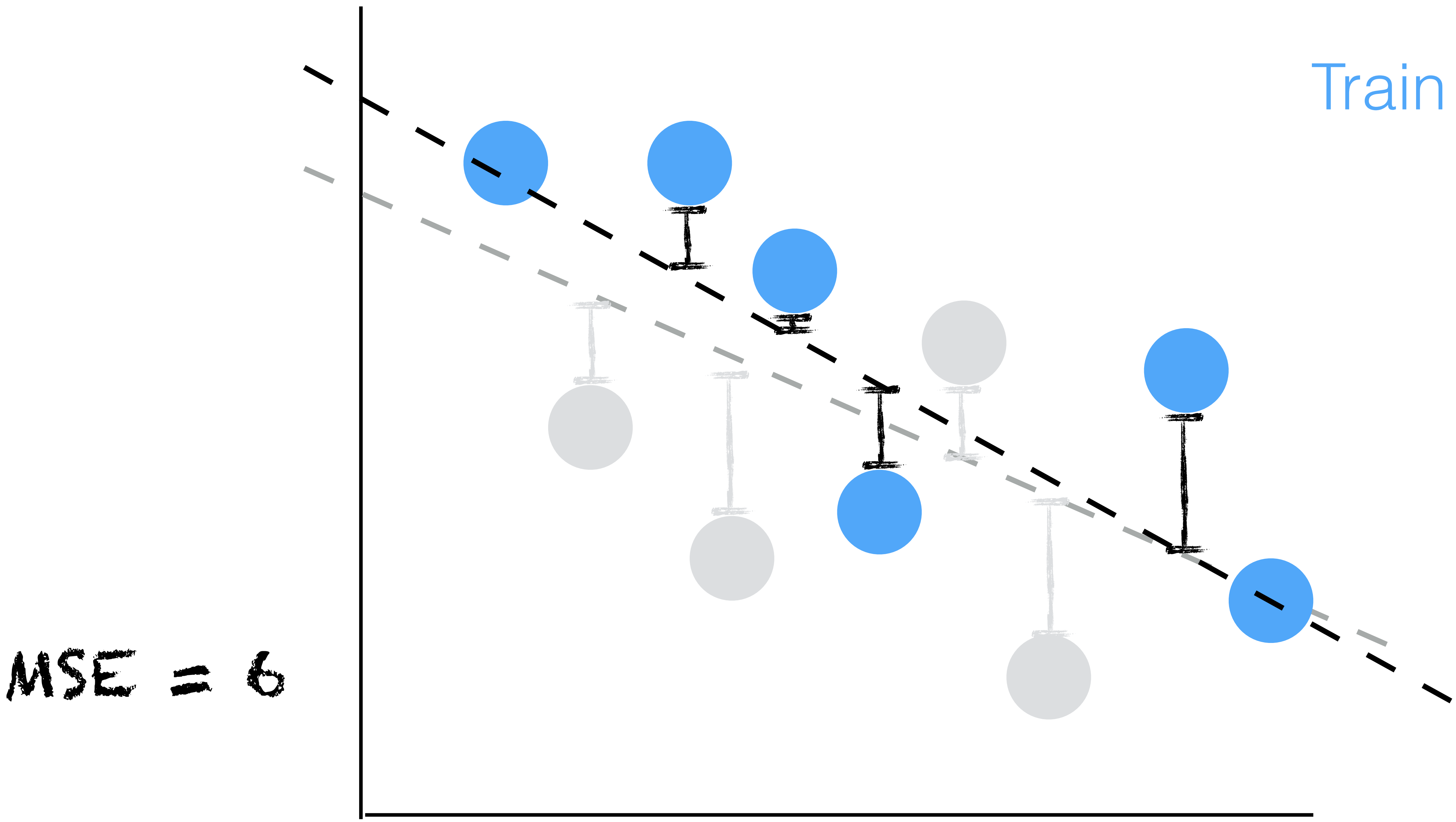
Train/Test Splits



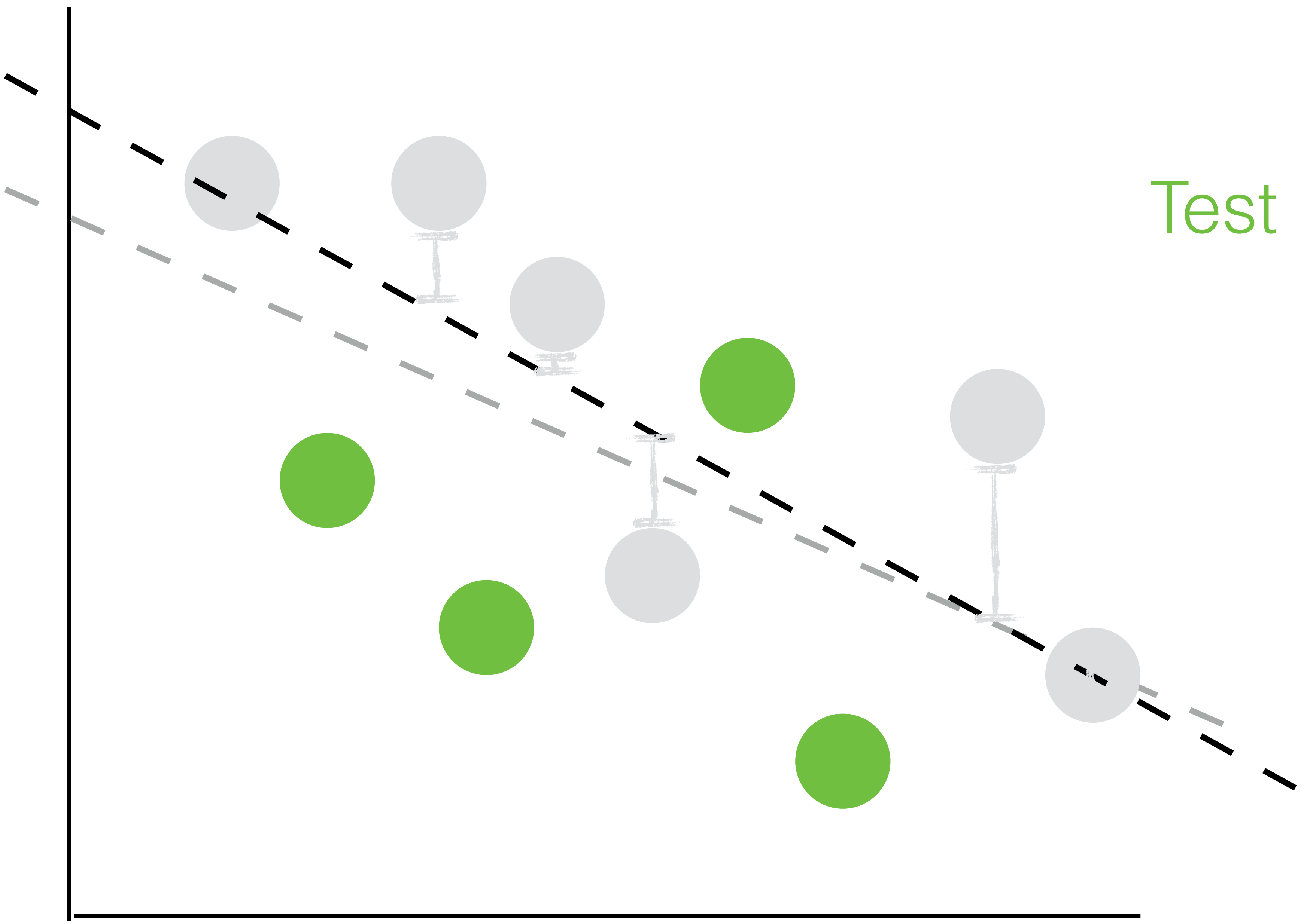
Train/Test Splits



Train/Test Splits



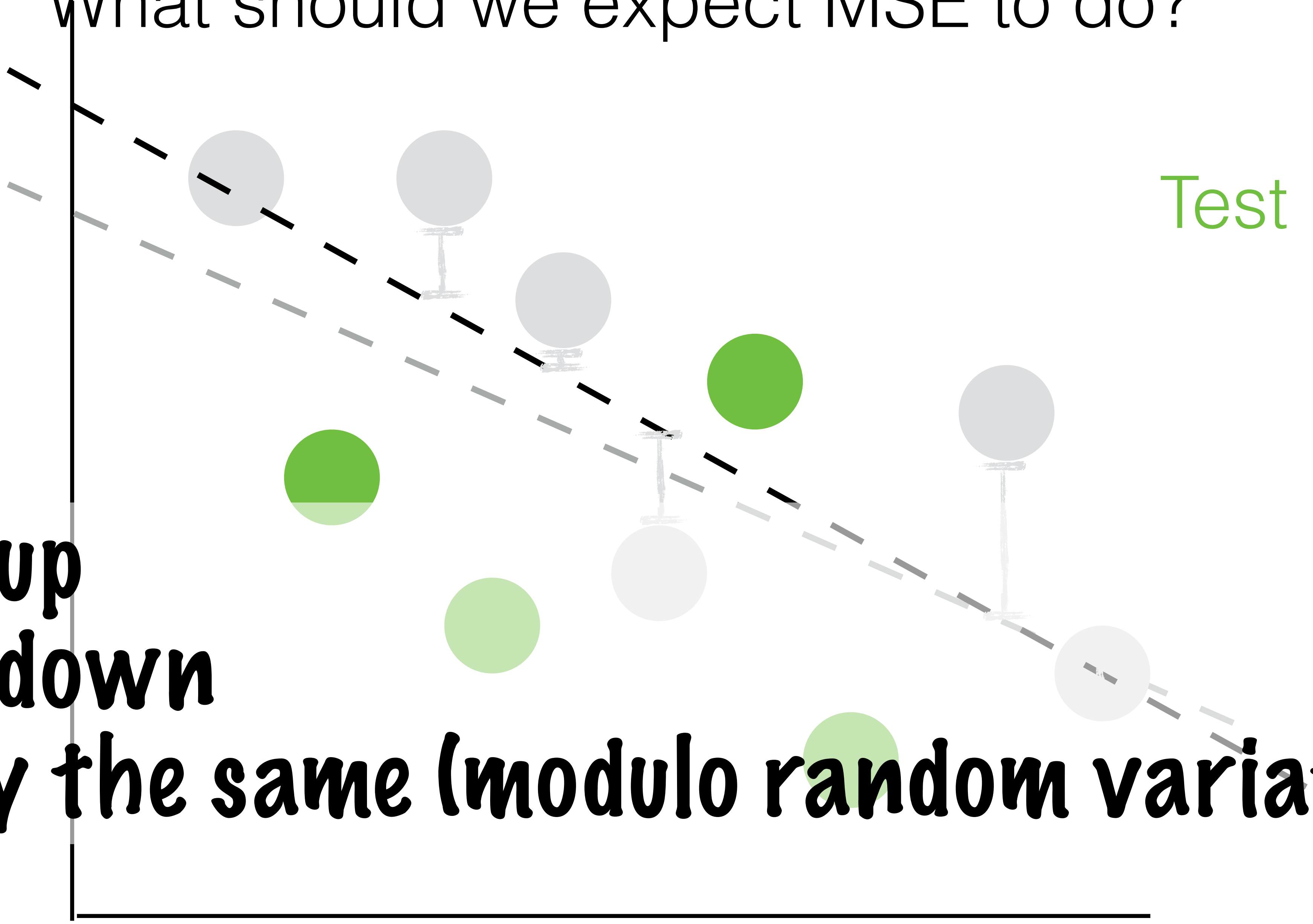
Train/Test Splits



Clicker Question!

Clicker Question!

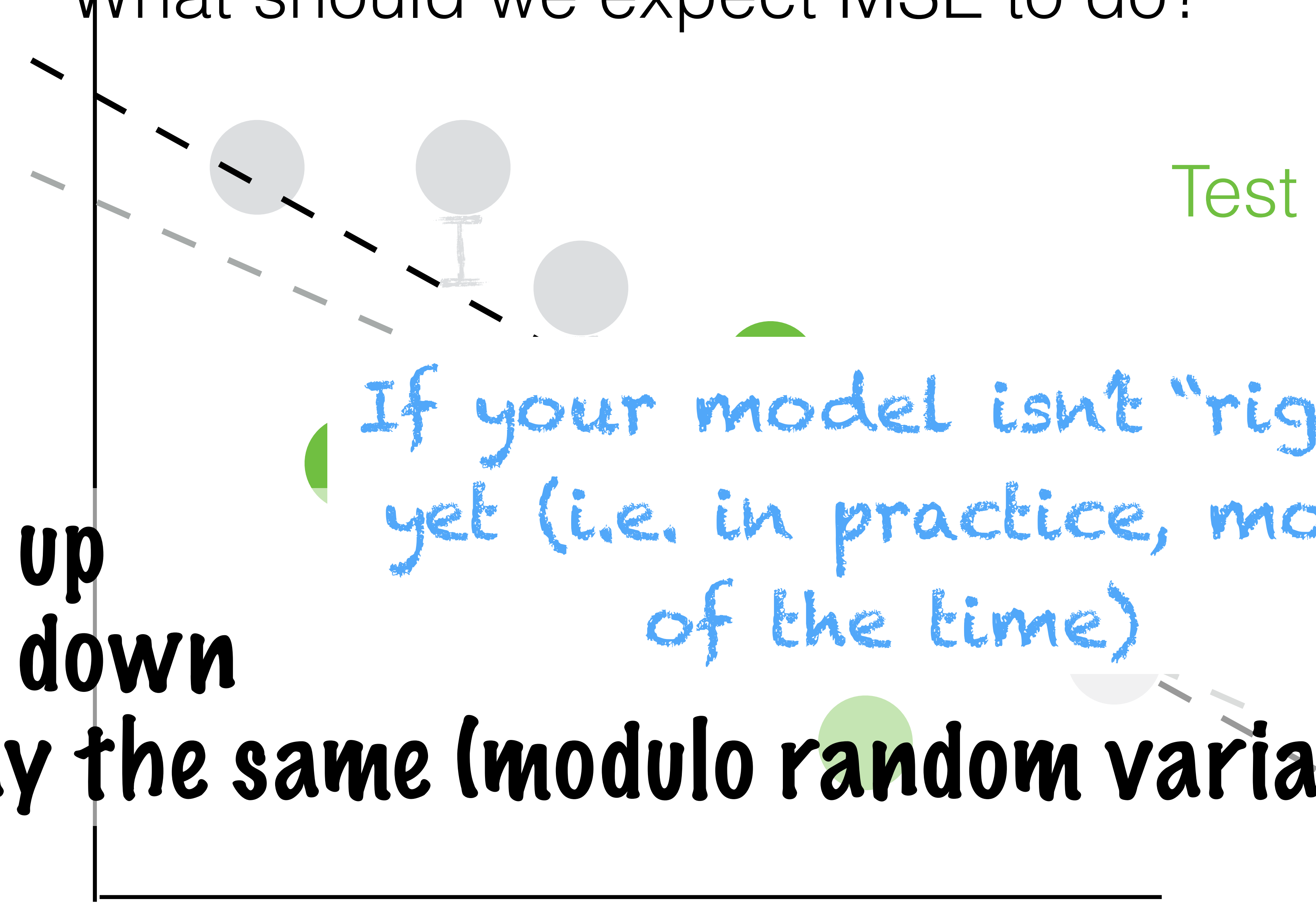
What should we expect MSE to do?



(a) Go up
(b) Go down
(c) Stay the same (modulo random variation)

Clicker Question!

What should we expect MSE to do?

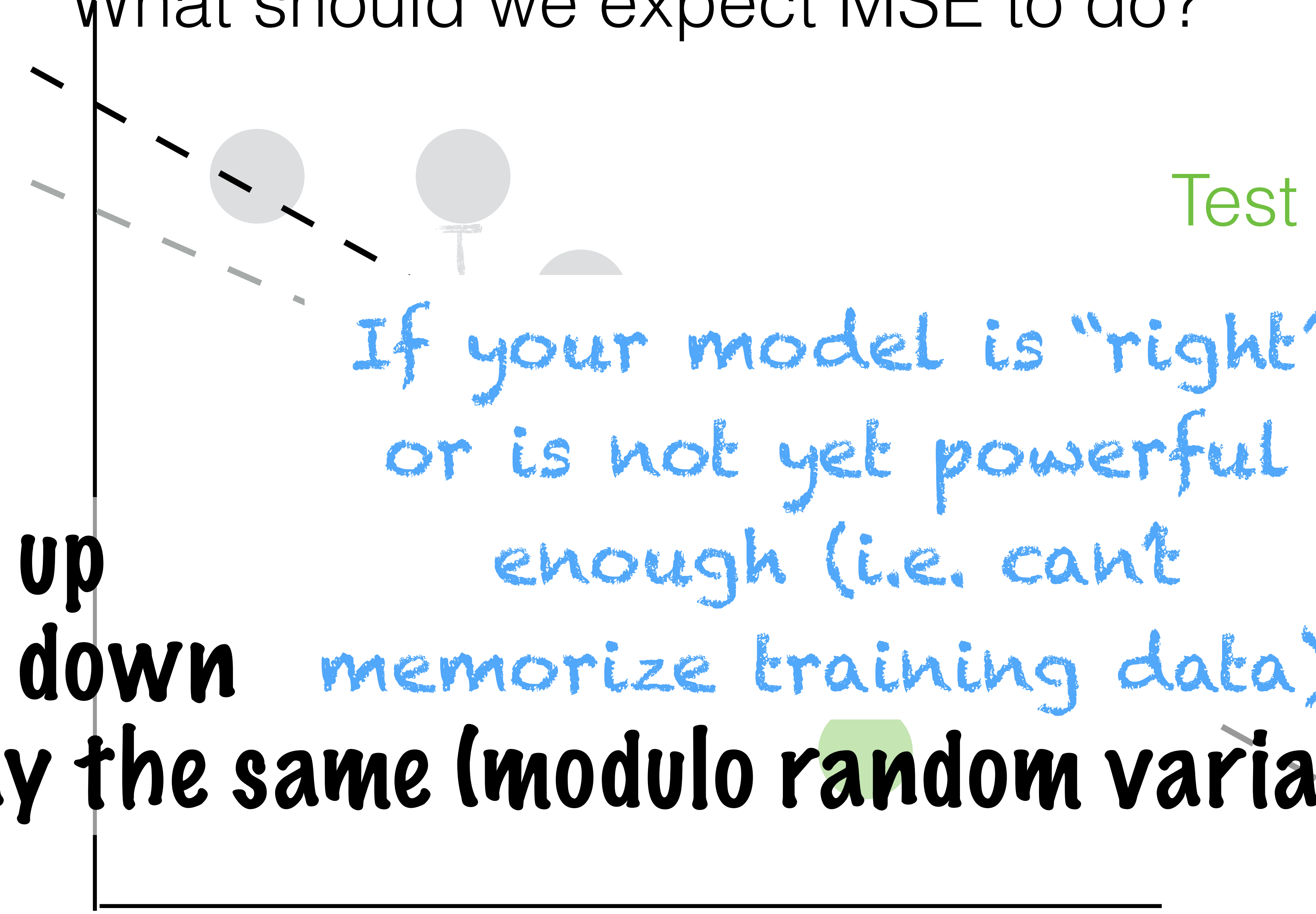


If your model isn't "right" yet (i.e. in practice, most of the time)

(a) Go up
(b) Go down
(c) Stay the same (modulo random variation)

Clicker Question!

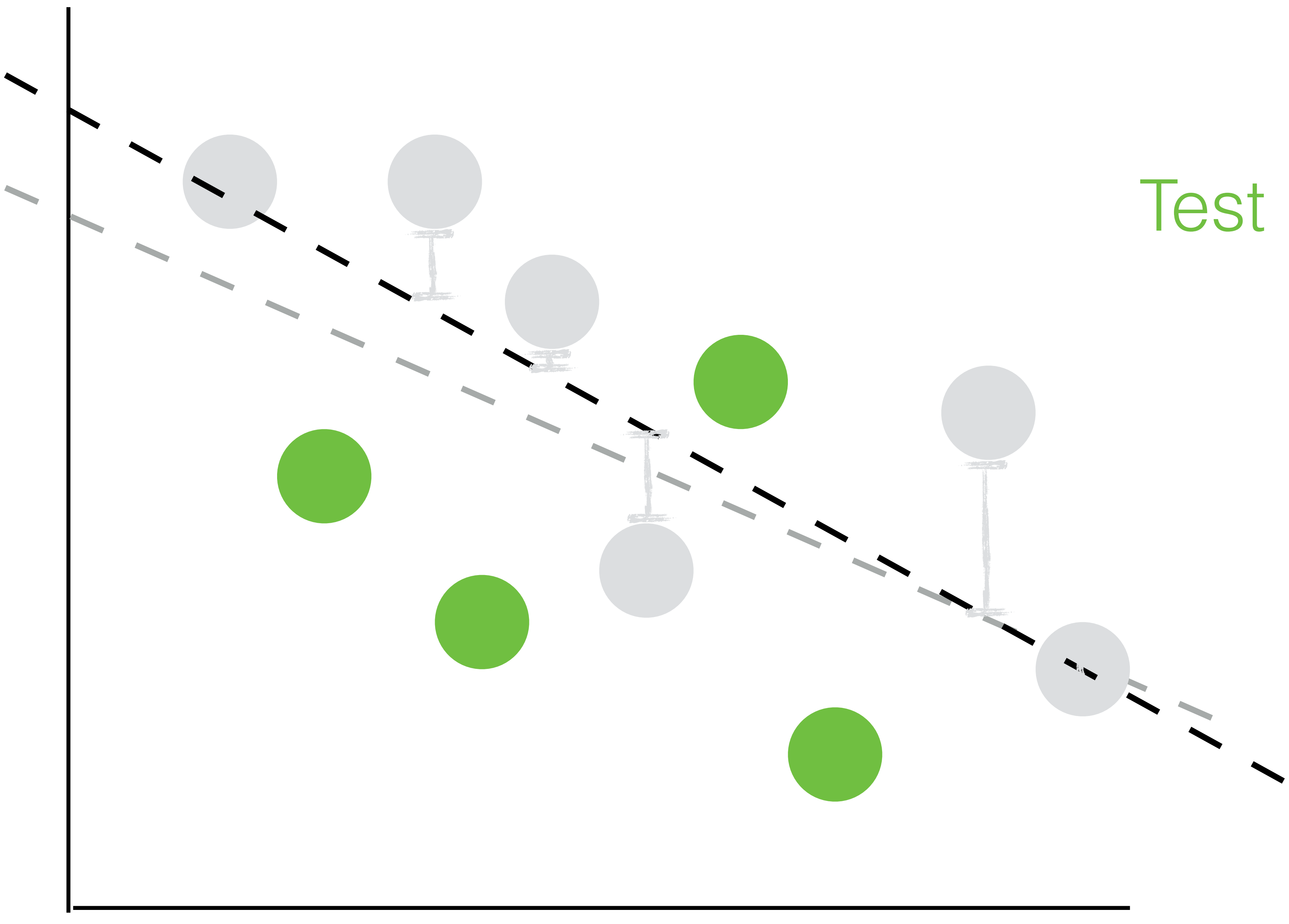
What should we expect MSE to do?



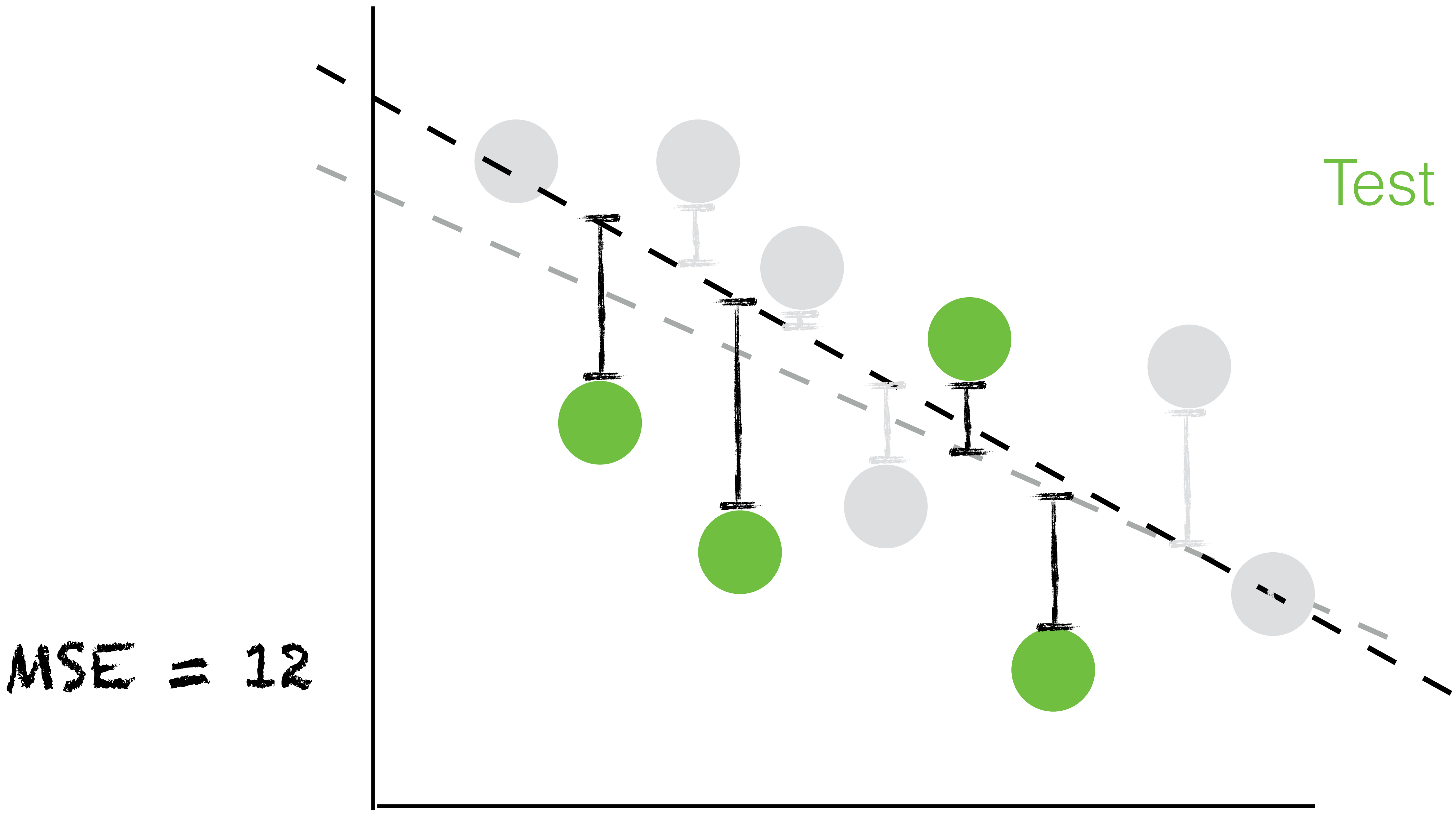
If your model is "right" or is not yet powerful enough (i.e. cant memorize training data).

(a) Go up
(b) Go down
(c) Stay the same (modulo random variation)

Train/Test Splits

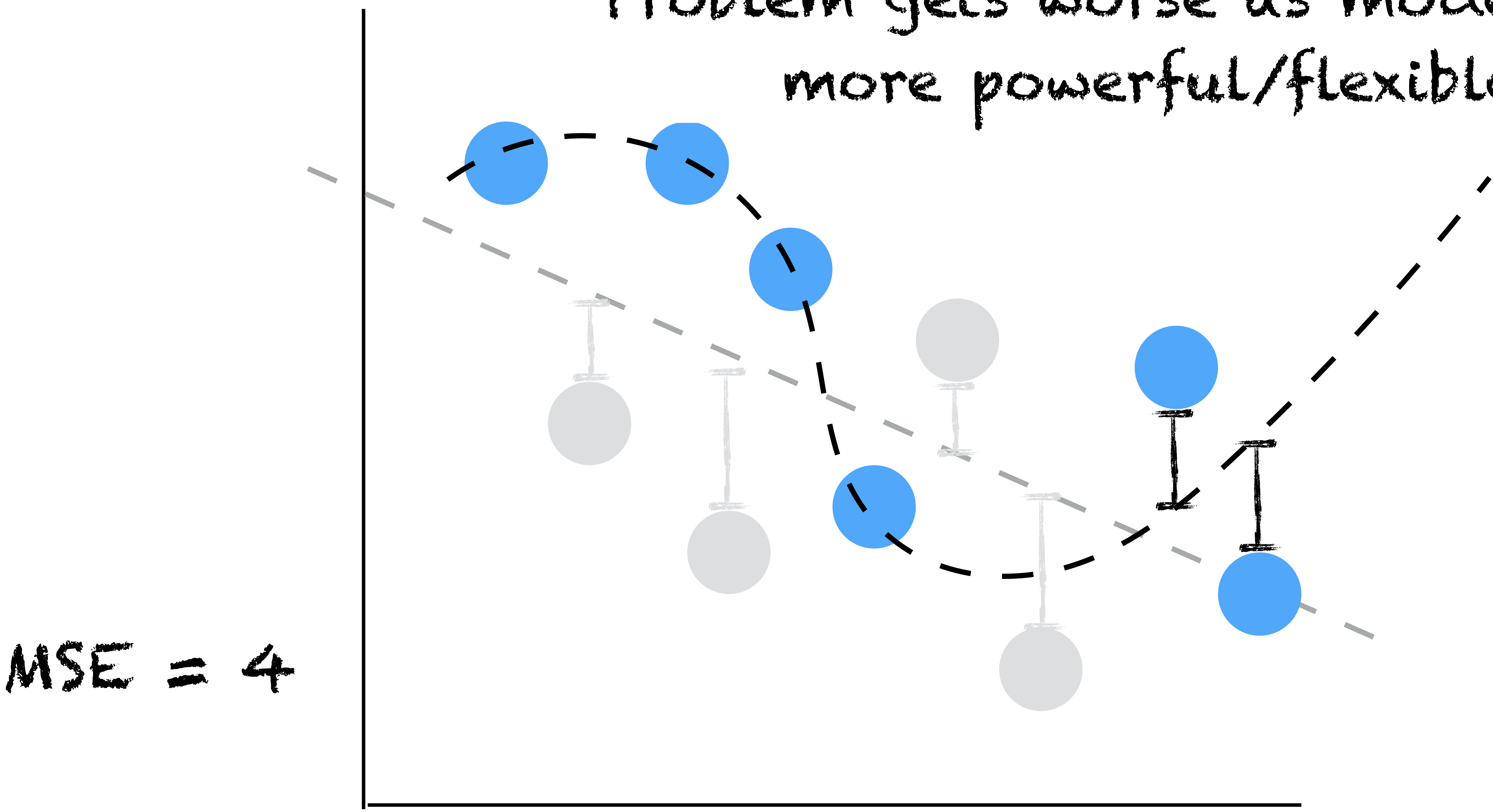


Train/Test Splits



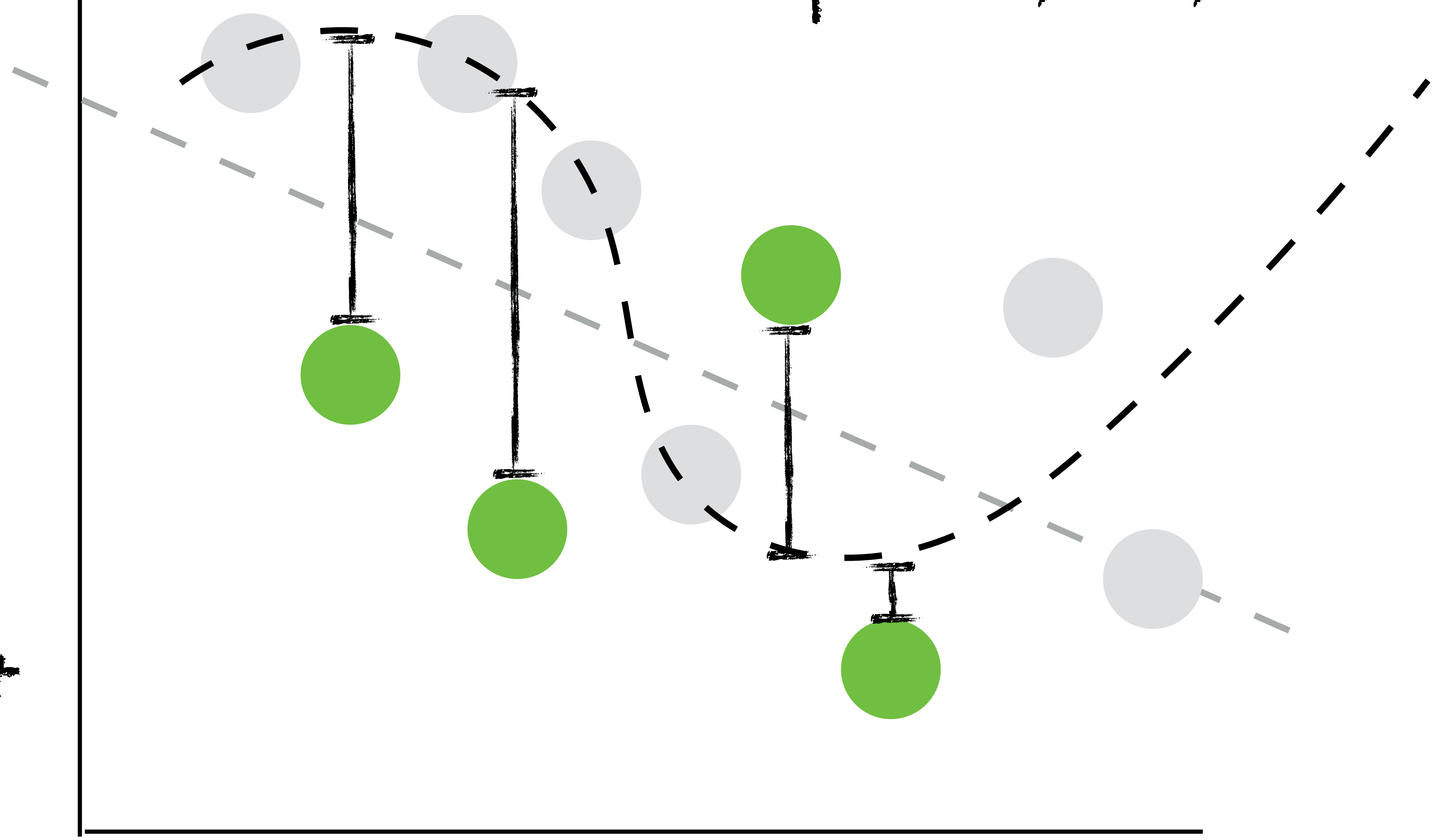
Train/Test Splits

Problem gets worse as models get more powerful/flexible



Train/Test Splits

Problem gets worse as models get more powerful/flexible



MSE = 14

Today

- ML "preliminaries"—terminology, basic building blocks, conceptual background
- **The two faces of linear regression**
- Training with Stochastic Gradient Descent

Regression Analysis in Stats	Regression in ML

Regression Analysis in Stats	Regression in ML
☑ Make claims about whether there is a meaningful relationship between X and Y	

Regression Analysis in Stats	Regression in ML
☑ Make claims about whether there is a meaningful relationship between X and Y	☑ Given X, predict Y; deploy a model to make predictions for new inputs

Regression Analysis in Stats	Regression in ML
☑ Make claims about whether there is a meaningful relationship between X and Y ☑ (Often) interested in causation; focus on controls and removing colinearity	☑ Given X, predict Y; deploy a model to make predictions for new inputs

Regression Analysis in Stats	Regression in ML
☑ Make claims about whether there is a meaningful relationship between X and Y	☑ Given X, predict Y; deploy a model to make predictions for new inputs
☑ (Often) interested in causation; focus on controls and removing colinearity	☑ Focused on prediction accuracy; exploiting correlation is totally fine

Regression Analysis in Stats	Regression in ML
☑ Make claims about whether there is a meaningful relationship between X and Y	☑ Given X, predict Y; deploy a model to make predictions for new inputs
☑ (Often) interested in causation; focus on controls and removing colinearity	☑ Focused on prediction accuracy; exploiting correlation is totally fine
☑ A “result” is typically in the form of a significant relationship and/or practically relevant effect size	

Regression Analysis in Stats

- Make claims about whether there is a meaningful relationship between X and Y
- (Often) interested in causation; focus on controls and removing colinearity
- A "result" is typically in the form of a significant relationship and/or practically relevant effect size

Regression in ML

- Given X, predict Y; deploy a model to make predictions for new inputs
- Focused on prediction accuracy; exploiting correlation is totally fine
- A "result" is typically in the form of an improvement in prediction performance on a (held out) test set

Regression Analysis in Stats	Regression in ML
☑ Make claims about whether there is a meaningful relationship between X and Y	☑ Given X, predict Y; deploy a model to make predictions for new inputs
☑ (Often) interested in causation; focus on controls and removing colinearity	☑ Focused on prediction accuracy; exploiting correlation is totally fine
☑ A “result” is typically in the form of a significant relationship and/or practically relevant effect size	☑ A “result” is typically in the form of an improvement in prediction performance on a (held out) test set
☑ Avoid overfitting by preferring simple models; avoid overclaiming by accounting for “degrees of freedom” when computing p values	

Regression Analysis in Stats	Regression in ML
Make claims about whether there is a meaningful relationship between X and Y	Given X, predict Y; deploy a model to make predictions for new inputs
(Often) interested in causation; focus on controls and removing colinearity	Focused on prediction accuracy; exploiting correlation is totally fine
A “result” is typically in the form of a significant relationship and/or practically relevant effect size	A “result” is typically in the form of an improvement in prediction performance on a (held out) test set
Avoid overfitting by preferring simple models; avoid overclaiming by accounting for “degrees of freedom” when computing p values	Avoid overfitting through regularization; avoid overclaiming by maintaining train/test splits and reporting test performance

Regression Analysis in Stats	Regression in ML
Make claims about whether there is a meaningful relationship between X and Y	Given X, predict Y; deploy a model to make predictions for new inputs
(Often causa and re	n ne
A "res form c relatio releva	n the form prediction ld out)
Avoid overfitting by preferring simple models; avoid overclaiming by accounting for "degrees of freedom" when computing p values	Avoid overfitting through regularization; avoid overclaiming by maintaining train/test splits and reporting test performance

But! These are the <u>same model.</u> These difference are "in general"/"by convention", not anything fundamental.

Regression Analysis in Stats	Regression in ML
☑ Make claims about whether there is a meaningful relationship between X and Y	☑ Given X, predict Y; deploy a model to make predictions for new inputs
☑ (Often) interested in causation; focus on controls and	☑ Focused on prediction accuracy; exploiting
☑ A "r form relationship and/or practically relevant effect size	☑ e form diction performance on a (held out) test set
☑ Avoid overfitting by preferring simple models; avoid overclaiming by accounting for "degrees of freedom" when computing p values	☑ Avoid overfitting through regularization; avoid overclaiming by maintaining train/test splits and reporting test performance

Different scientific communities with different goals.

Regression Analysis in Stats	Regression in ML
Make claims about whether there is a meaningful relationship between X and Y	Given X, predict Y; deploy a model to make predictions for new inputs
(Oft cau and	
A "r forr rela rele	e form diction ut)
Avoid overfitting by preferring simple models; avoid overclaiming by accounting for "degrees of freedom" when computing p values	Avoid overfitting through regularization; avoid overclaiming by maintaining train/test splits and reporting test performance

Different scientific communities with different goals.

(and different software packages :))

<- R, stats_models, STATA

sklearn, matlab, pytorch ->

Regression Analysis in Stats	Regression in ML
☑ Make claims about whether there is a meaningful relationship between X and Y	☑ Given X, predict Y; deploy a model to make predictions for new inputs
☑ (Often) interested in causation; focus on controls and removing colinearity	☑ Focused on prediction accuracy; exploiting correlation is totally fine
☑ A “result” is typically in the form of a significant relationship and/or practically relevant effect size	☑ A “result” is typically in the form of an improvement in prediction performance on a (held out) test set
preferring id counting dom” values	☑ Avoid overfitting through regularization; avoid overclaiming by maintaining train/test splits and reporting test performance

In the limit, I think these goals *are* the same. Even if we care about prediction (and we want to do it using as few models as possible), shouldn't we get the best performance by modeling the “true” underlying process?

Isn't it the case that correct explanatory/causal models necessarily make right predictions, but not vice-versa?

☑ Avoid overfitting through regularization; avoid overclaiming by maintaining train/test splits and reporting test performance

Counter argument: You can get perfect* predictive performance with the wrong model. We were extremely good at predicting whether objects would fall or float long before we knew about gravity.

Explanatory/causal models are hard! We might never get there. Maybe empirically accurate predictions should lead, and theory/explanation will follow?

- form... relationsh... relevant effect size
- Avoid overfitting by preferring simple models; avoid overclaiming by accounting for "degrees of freedom" when computing p values

...m ...diction ...held out)

- Avo... reg... ove... trai... tes...

Today

- ML “preliminaries”—terminology, basic building blocks, conceptual background
- The two faces of linear regression
- **Training with Stochastic Gradient Descent**

Model

#1

- Make assumptions about the problem domain.
- How is the data generated?
- How is the decision-making procedure structured?
- What types of dependencies exist?
- Trending buzzword: “inductive biases”

#2

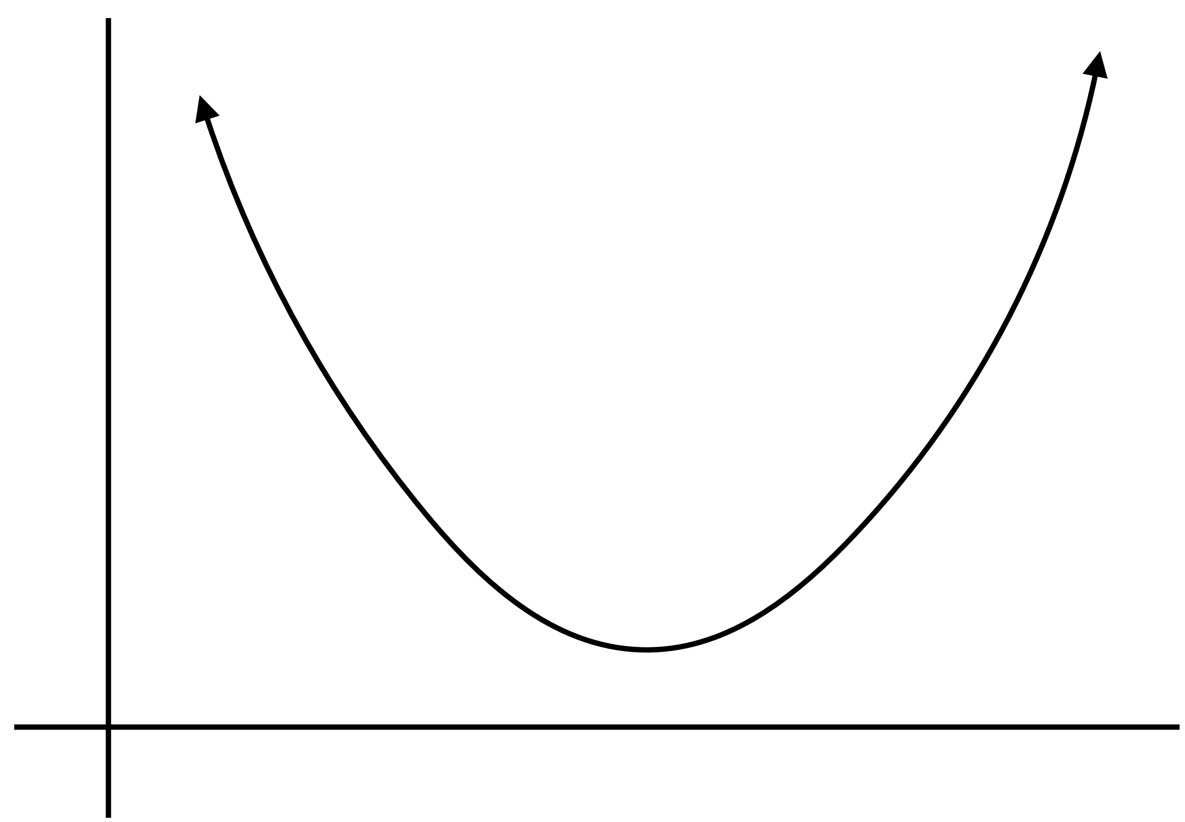
- How to train the model?

Training with Gradient Descent

minimize $$\sum_{i=1}^{n}(Y_i - \hat{Y})^2$$

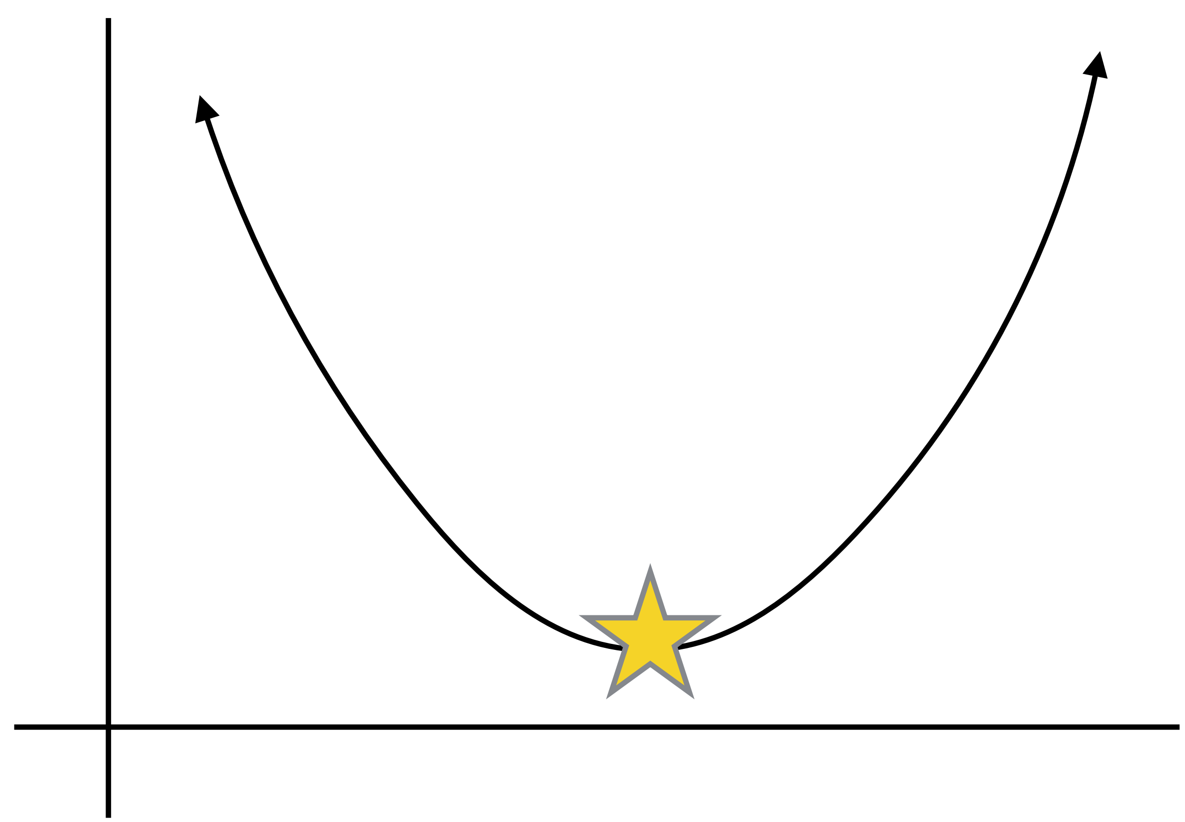
Training with Gradient Descent

minimize $$\sum_{i=1}^{n}(Y_i - \hat{Y})^2$$



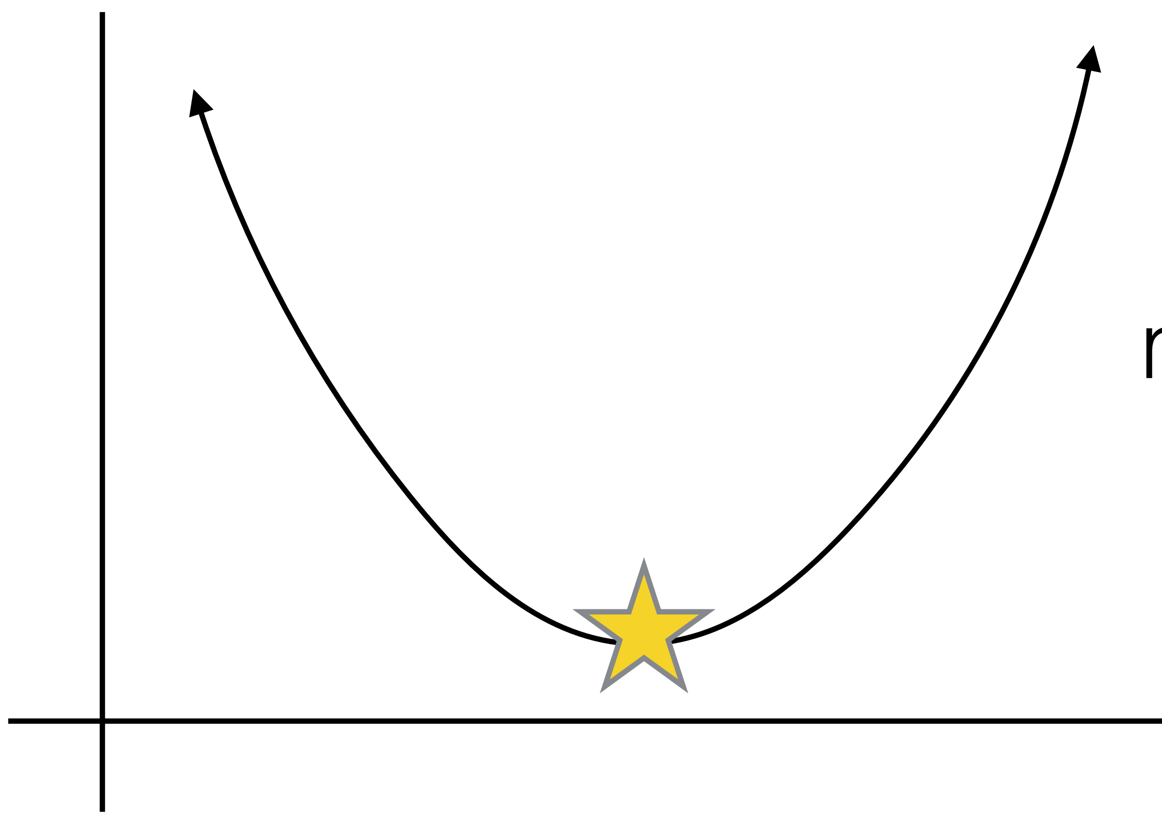
Training with Gradient Descent

minimize $$\sum_{i=1}^{n}(Y_i - \hat{Y})^2$$



Training with Gradient Descent

minimize $$\sum_{i=1}^{n}(Y_i - \hat{Y})^2$$

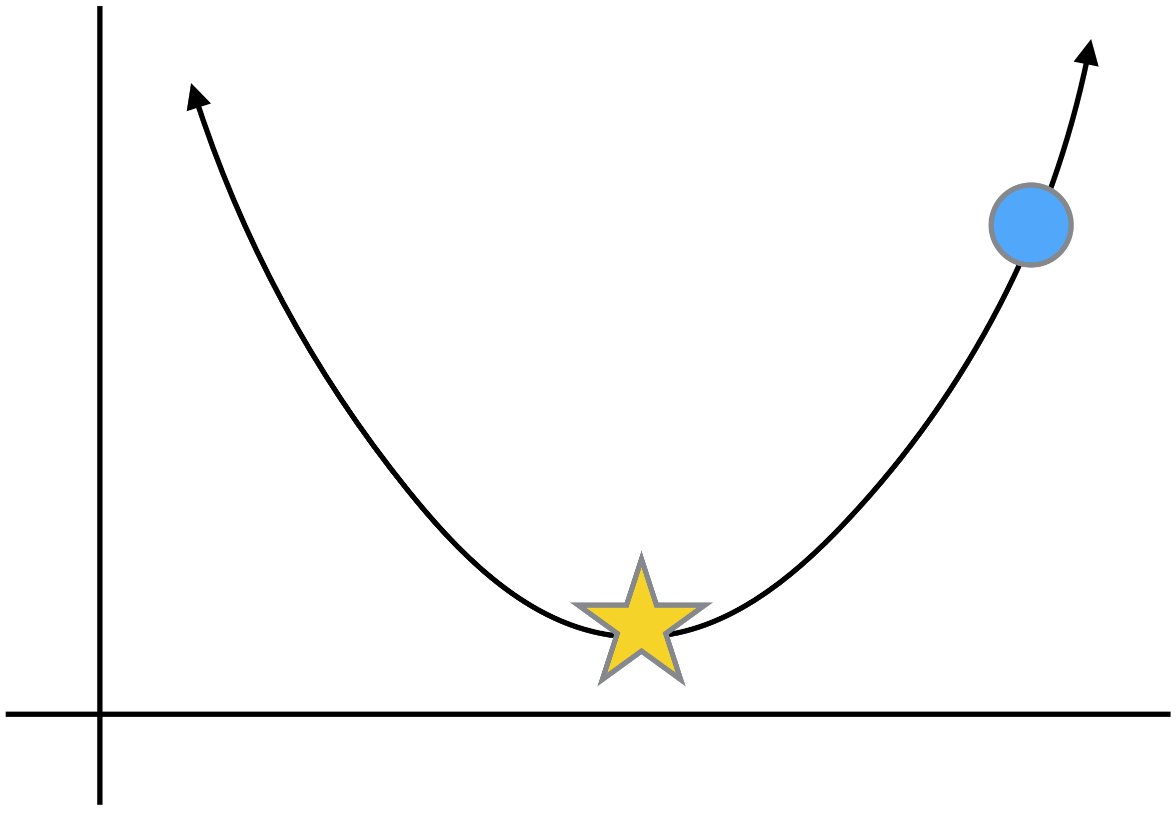


b = $\bar{Y} - m\bar{X}$

m = $\dfrac{Cov(X,Y)}{Var(X)}$

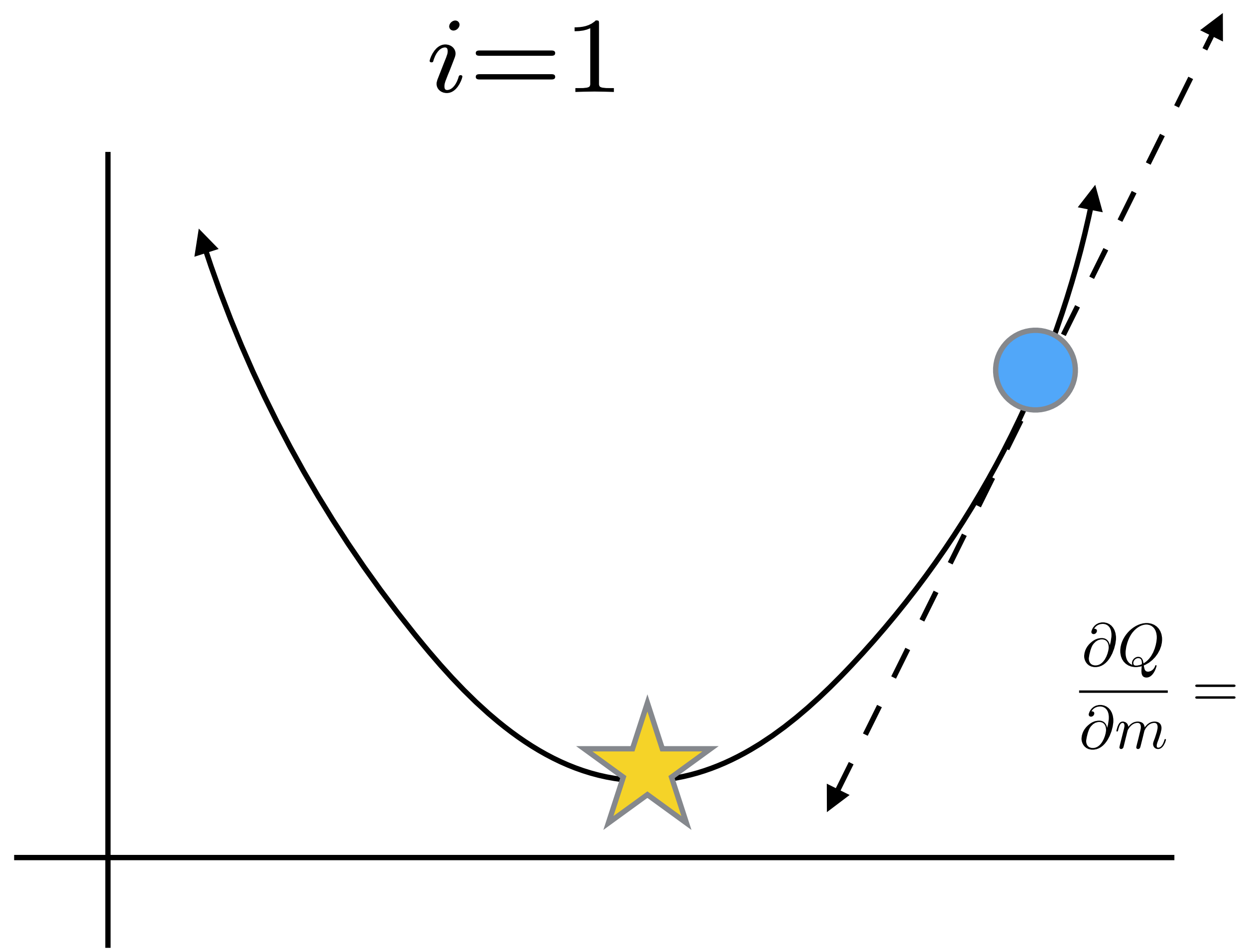
Training with Gradient Descent

minimize $$\sum_{i=1}^{n}(Y_i - \hat{Y})^2$$



Training with Gradient Descent

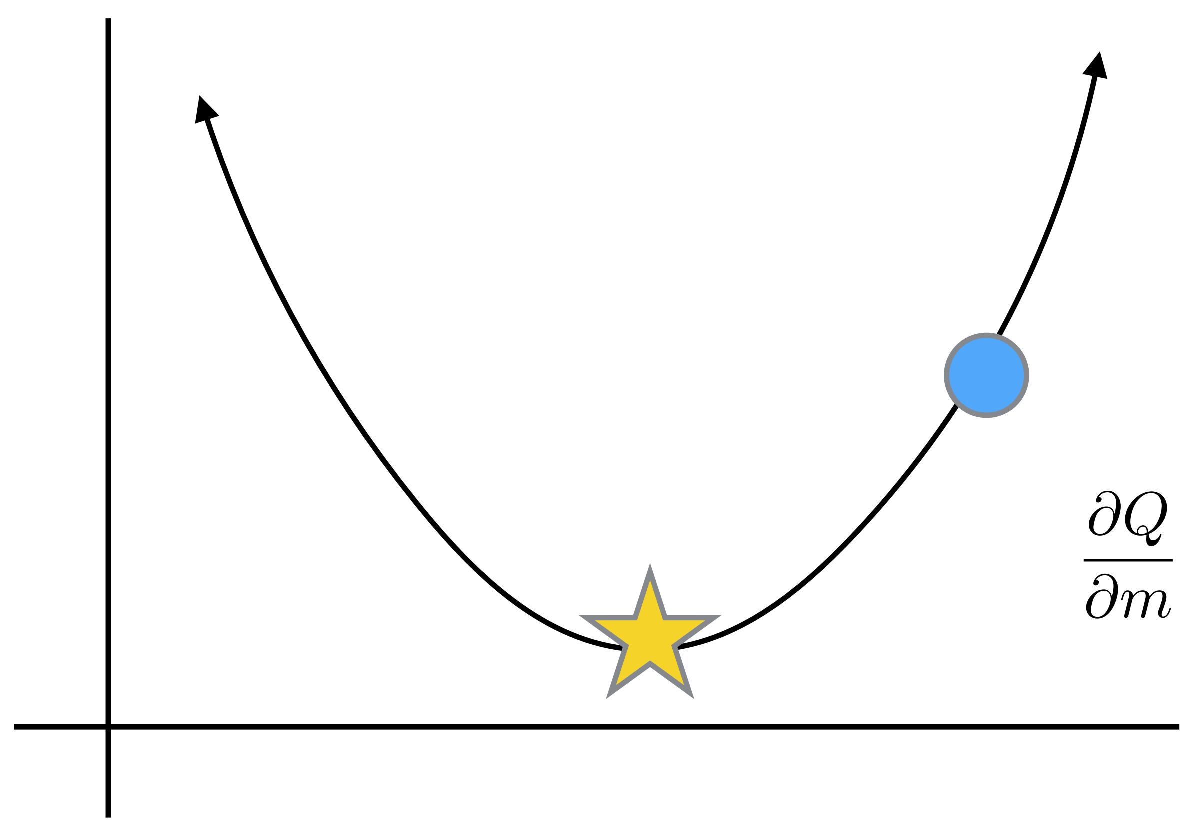
minimize $$\sum_{i=1}^{n}(Y_i - \hat{Y})^2$$



$$\frac{\partial Q}{\partial m} = \sum_{i=1}^{n} -2X_i(Y_i - b - mX_i)$$

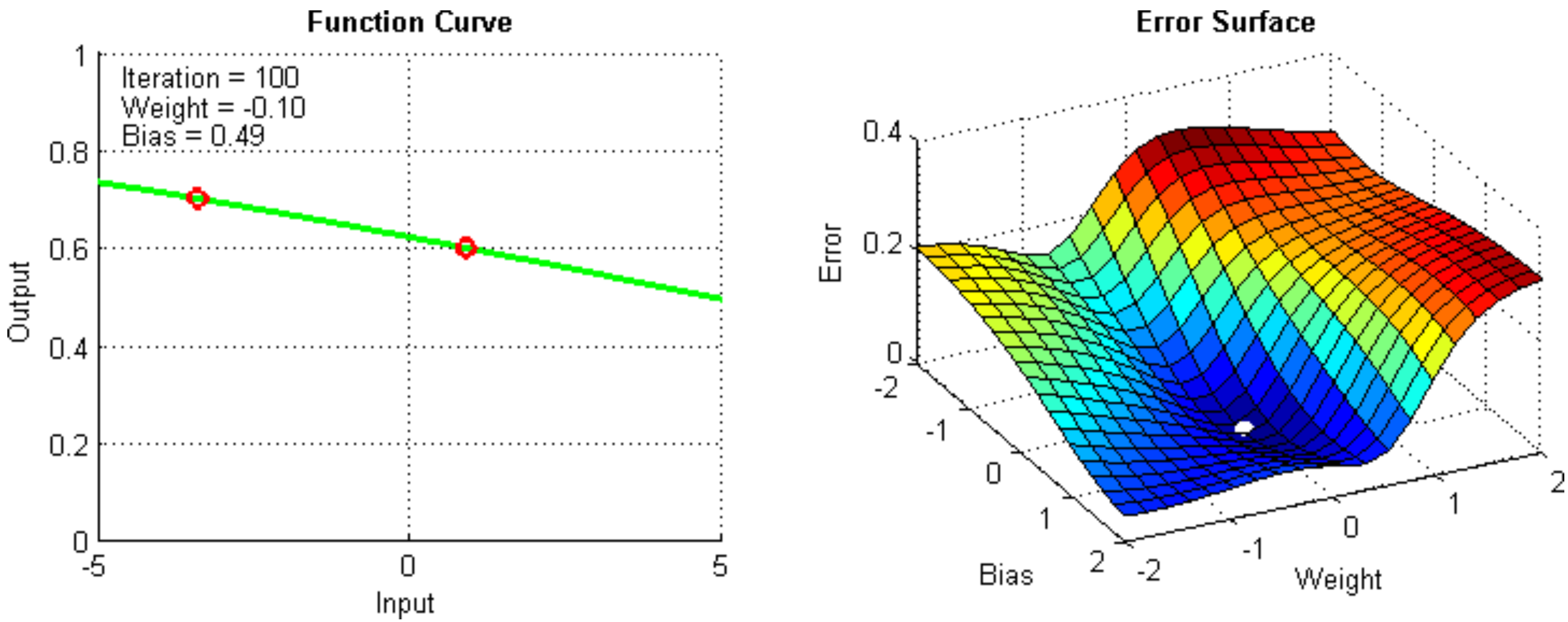
Training with Gradient Descent

minimize $$\sum_{i=1}^{n}(Y_i - \hat{Y})^2$$



$$\frac{\partial Q}{\partial m} = \sum_{i=1}^{n} -2X_i(Y_i - b - mX_i)$$

Training with Gradient Descent



https://independentseminarblog.com/2018/01/12/moving-below-the-surface-3-gradient-descent-william/

Training with Gradient Descent

Helpful equations for following along in the jupyter notebook

$$Q = \sum_{i=1}^{n} (Y_i - (mX_i + b))^2$$

$$\frac{\partial Q}{\partial b} = \sum_{i=1}^{n} -2(Y_i - mX_i - b) = 0$$

$$\frac{\partial Q}{\partial m} = \sum_{i=1}^{n} -2X_i(Y_i - b - mX_i) = 0$$

$$m = \frac{Cov(X,Y)}{Var(X)} \qquad b = \bar{Y} - m\bar{X}$$

https://independentseminarblog.com/2018/01/12/moving-below-the-surface-3-gradient-descent-william/

👋