LECTURE 16

Feature Engineering

Transforming numerical features and encoding categorical data in order to build more sophisticated models.

Data 100/Data 200, Fall 2021 @ UC Berkeley

Fernando Pérez and Alvin Wan
(content by Alvin Wan, John DeNero, Joseph E. Gonzalez, Josh Hug)

LECTURE 16

Where a Linear Model Struggles

Understanding the downfalls of a linear model and where feature engineering is needed

Data 100/Data 200, Fall 2021 @ UC Berkeley

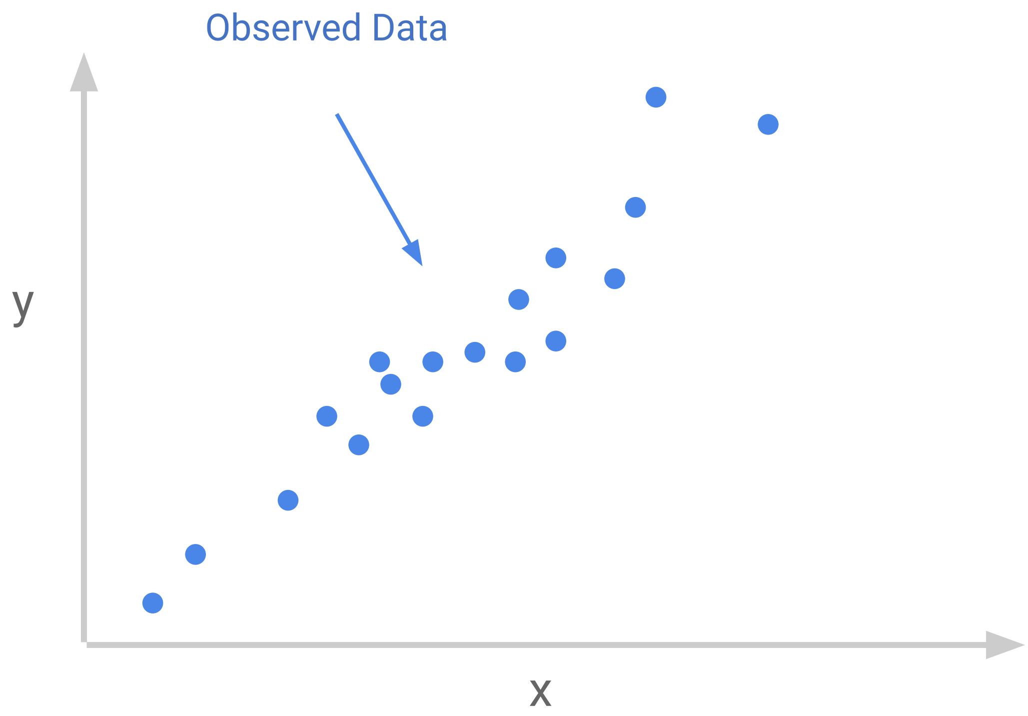
Fernando Pérez and Alvin Wan

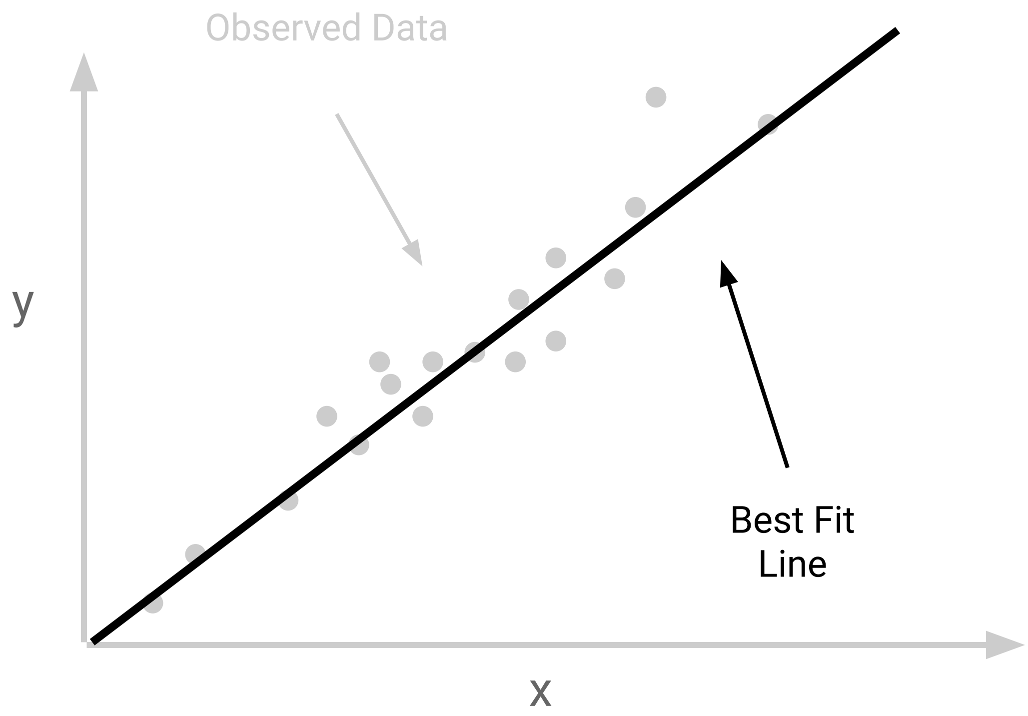
(content by Alvin Wan, John DeNero, Joseph E. Gonzalez, Josh Hug)

Recap Linear Models

Feature Engineering

Feature Functions

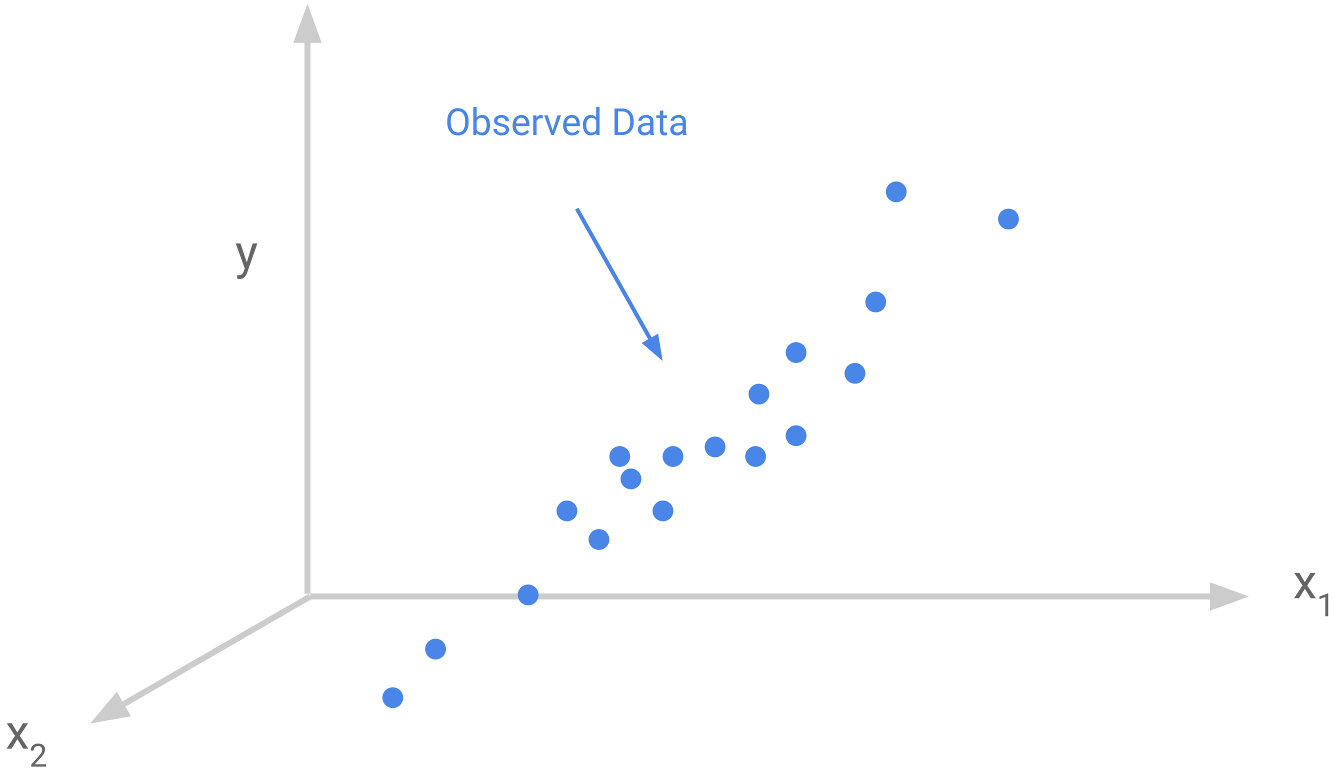

Observed Data
y
x


Observed Data
y
x
Best Fit
Line

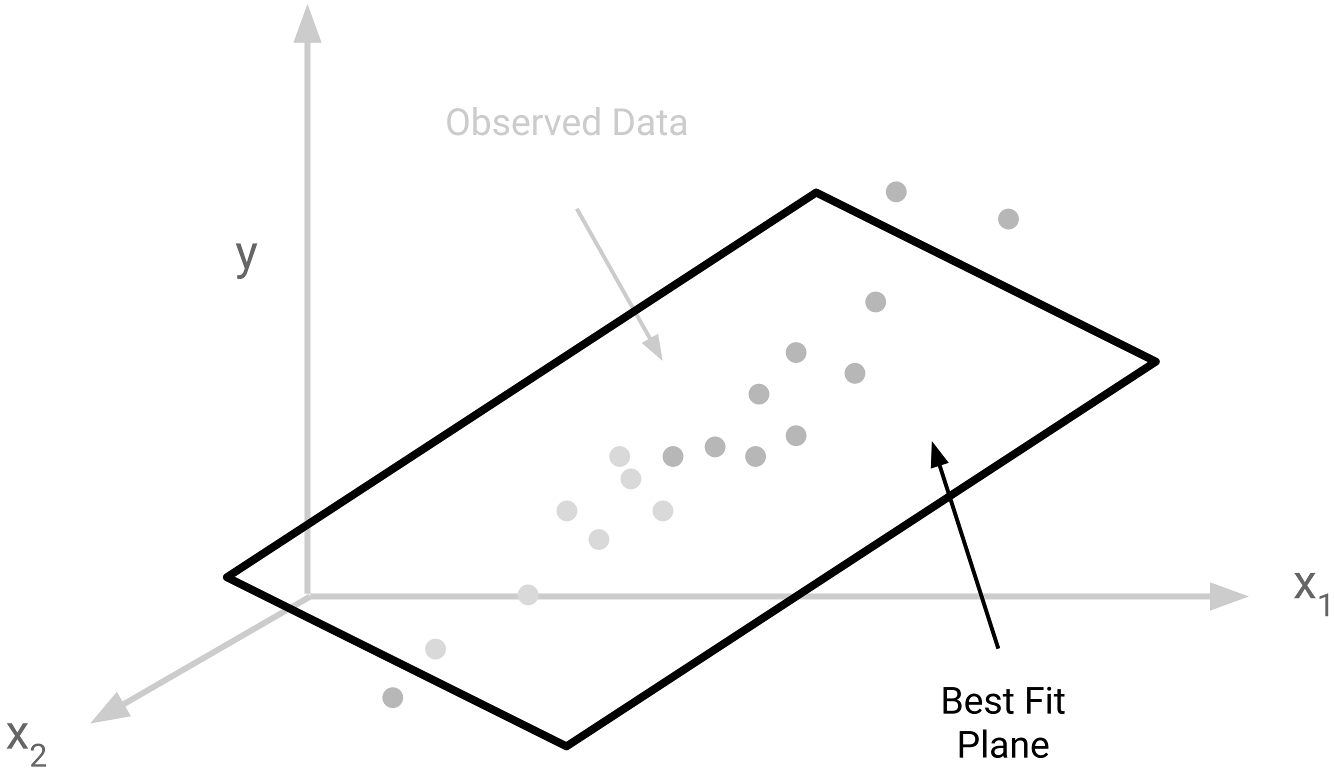
Parameters

Features

$$\hat{y} = x\theta$$



Observed Data
y
x1
x2


Observed Data
y
x_1
x_2
Best Fit Plane

Parameters

Features

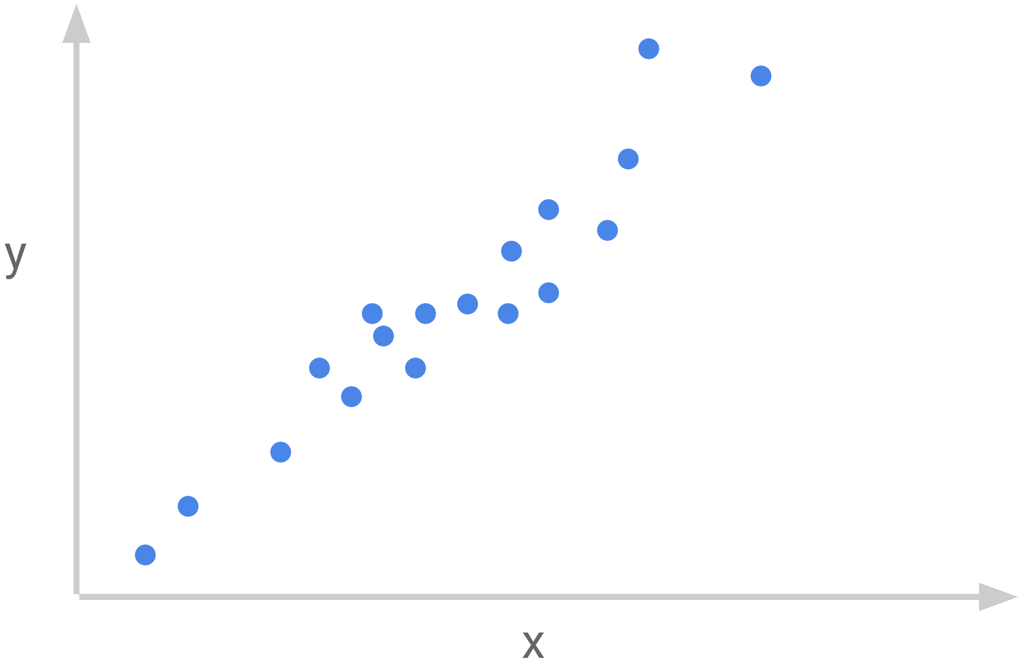
$$\hat{y} = x_1\theta_1 + x_2\theta_2$$

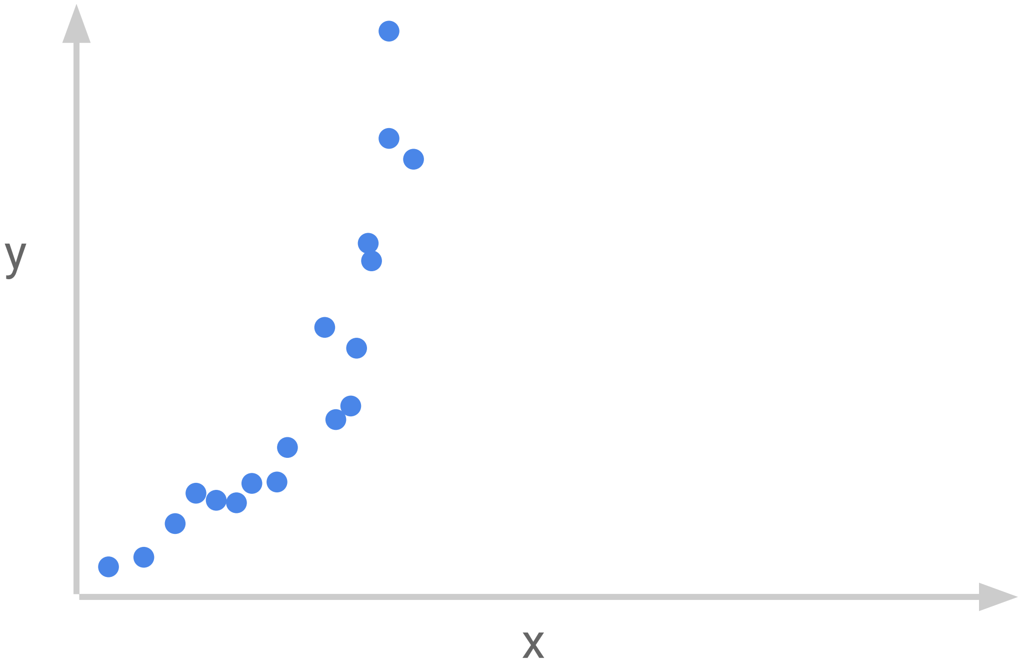
Rewrite

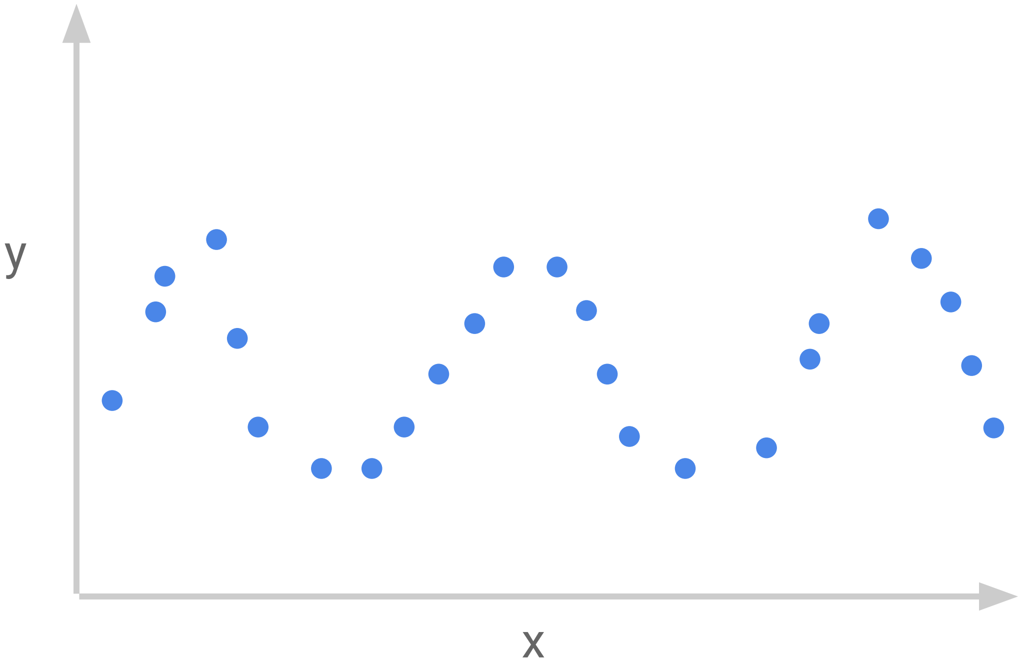
Parameters

Features

$$\hat{y} = \sum_{j=1}^{d} x_j \theta_j$$


y
x


y
x


y
x

What if the relationship between y and x_i is not linear?

What if the relationship between y and x_i is not linear?

What if our features x_i are not numbers?

Recap Linear Models

Feature Engineering

Feature Functions

TAKEAWAY

Feature engineering transforms **raw** features into more **informative** features for modeling.

TAKEAWAY

Feature engineering enables you to **express non-linear relationships**, **capture domain knowledge**, and **encode non-numeric features**.

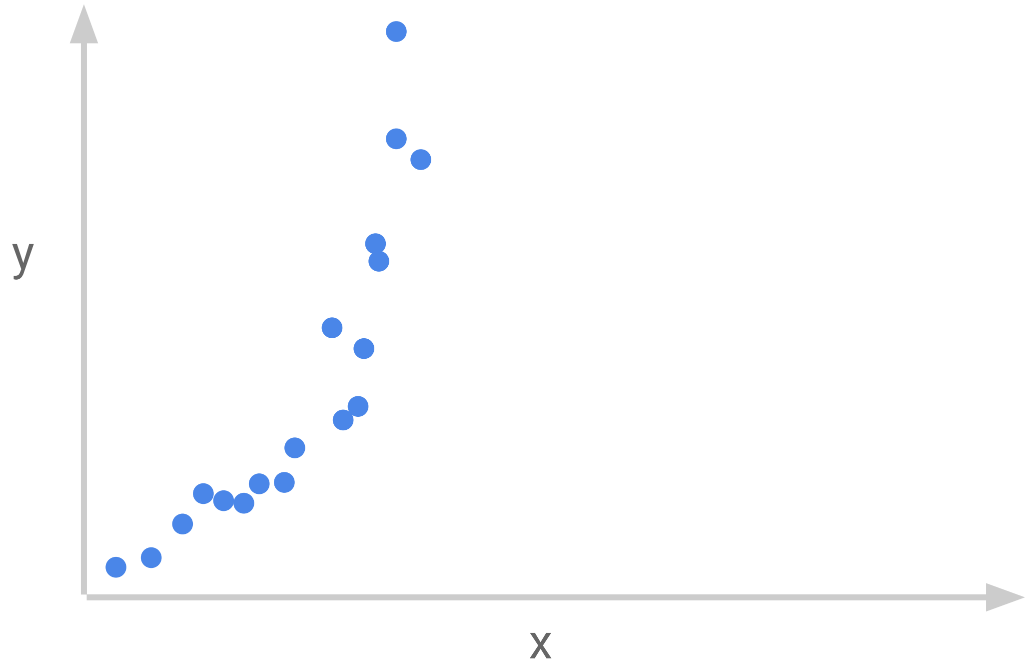
Recap Linear Models

Feature Engineering

Feature Functions

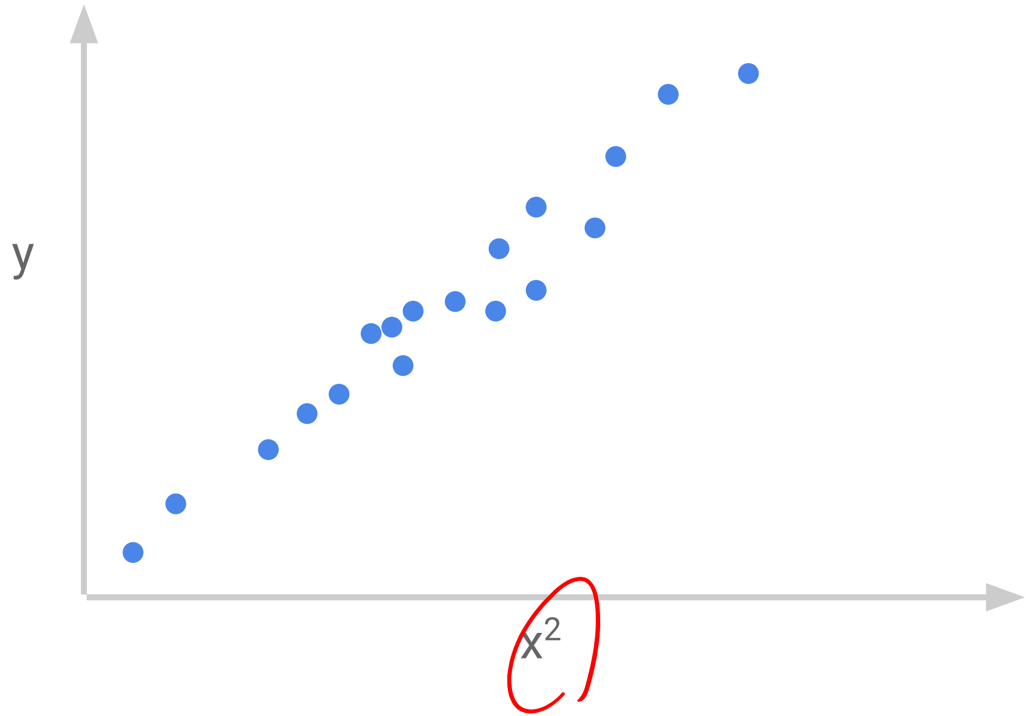
Non-linear relationship

between x and y



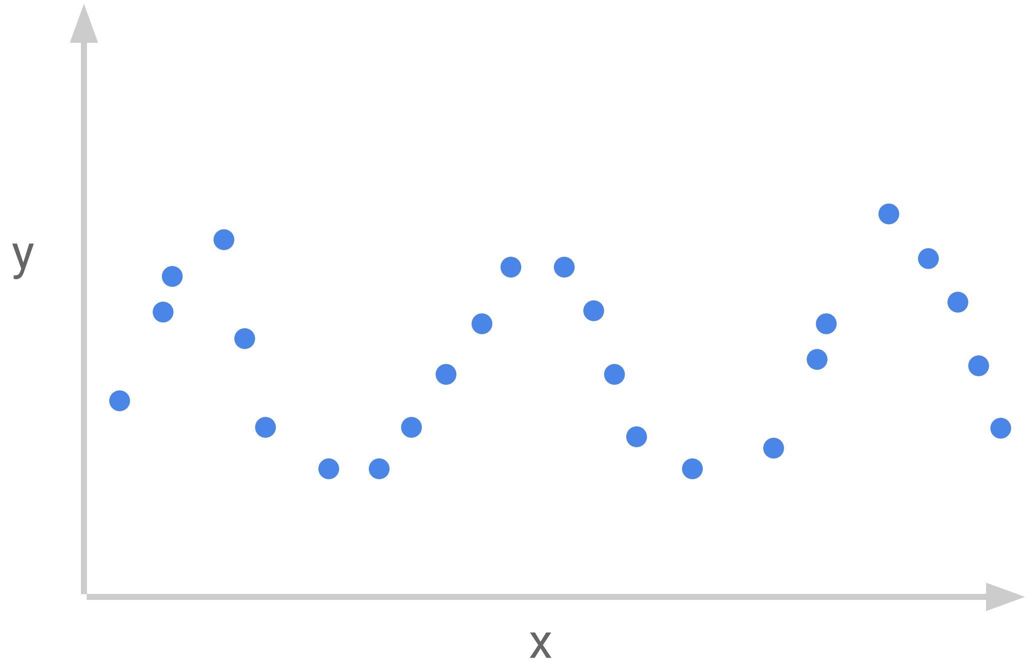
Linear relationship

between x^2 and y



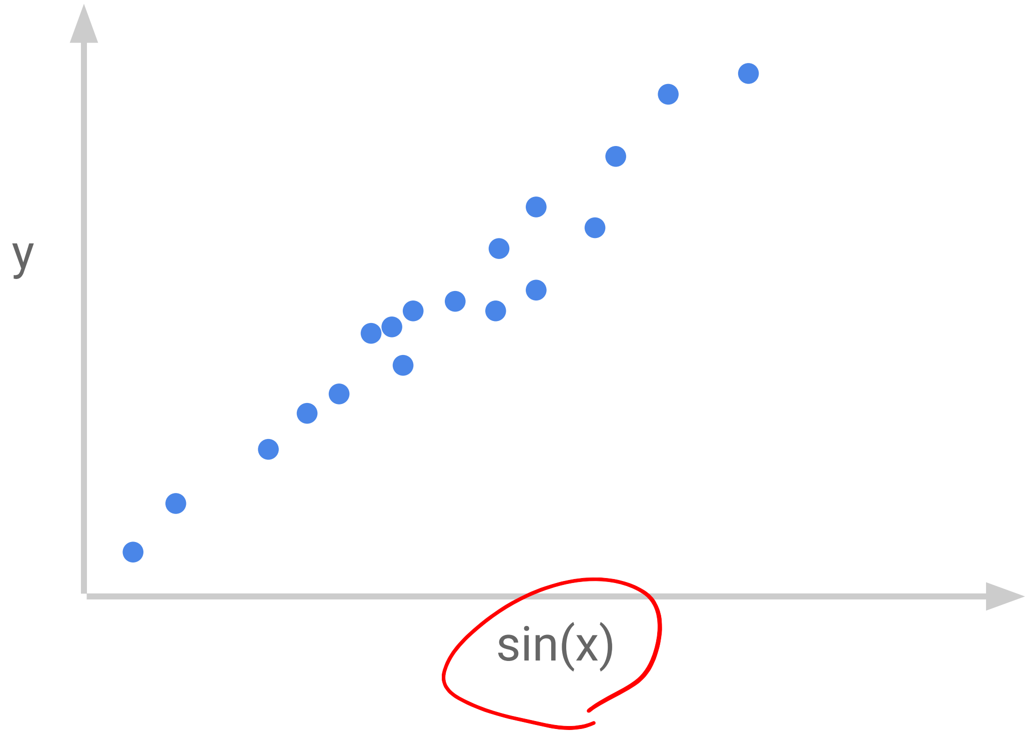
Non-linear relationship

between x and y



Linear relationship

between sin(x) and y



$$\hat{y} = \sum_{j=1}^{d} \overbrace{x_j}^{\text{d Features}} \underbrace{\theta_j}_{\text{Parameters}}$$

$$\hat{y} = \sum_{j=1}^{p} \overbrace{\underbrace{\phi}_{\text{(phi)ture function}}(x)_j}^{\text{p Features}} \underbrace{\theta_j}_{\text{Parameters}}$$

(phi)ture function
- Prof. Gonzalez, Sp21

new dimentional space

$$\phi : \mathbb{R}^d \rightarrow \mathbb{R}^p$$

Feature Engineering

Non-numeric and **Raw** Values

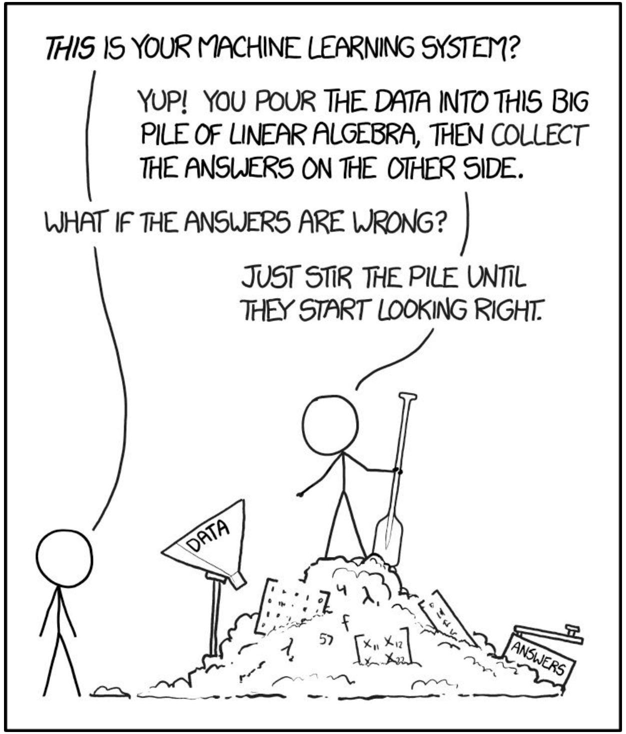
uid	age	state	hasBought	review
0	32	NY	True	"Meh."
42	50	WA	True	"Worked out of the box …"
57	16	CA	NULL	"Hella tots lit…"

$$\phi \Big\downarrow$$

Entirely **Quantitative** and **Transformed** Values

AK	…	NY	…	WY	age	age^2	hasBought missing
0	…	1	…	0	32	32^2	0
0	…	0	…	0	50	50^2	0
0	…	0	…	0	16	16^2	1

Designing feature functions is a big part of machine learning and data science.



xkcd.com/1838/

np.book

np. book

LECTURE 16

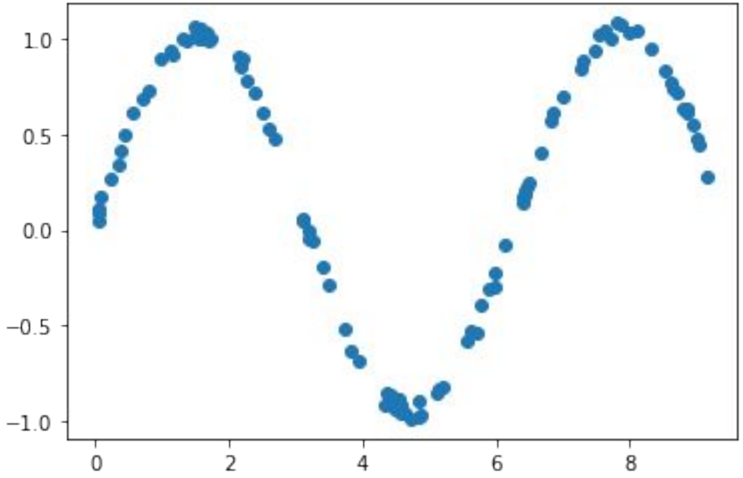
Conclusion

Gotchas for feature engineering. Warnings and pitfalls to be aware of.

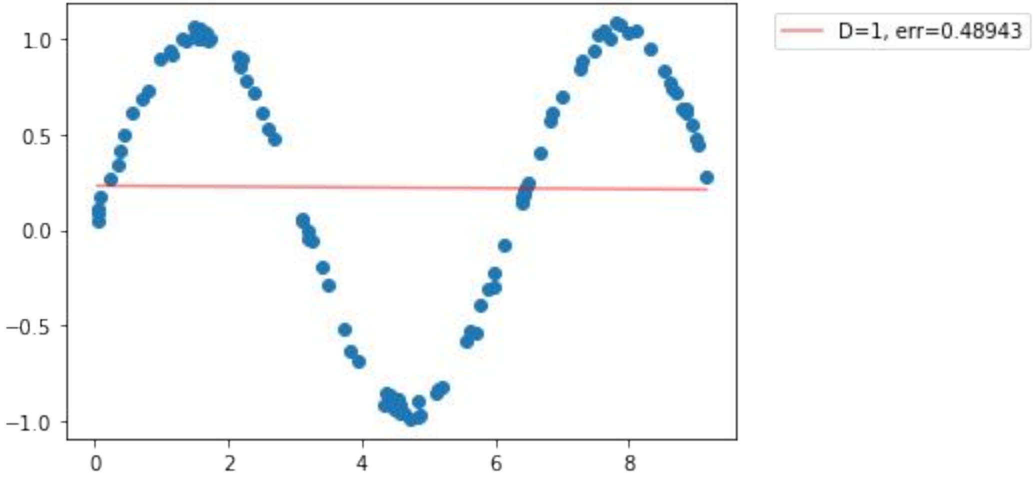
Data 100/Data 200, Fall 2021 @ UC Berkeley

Fernando Pérez and Alvin Wan

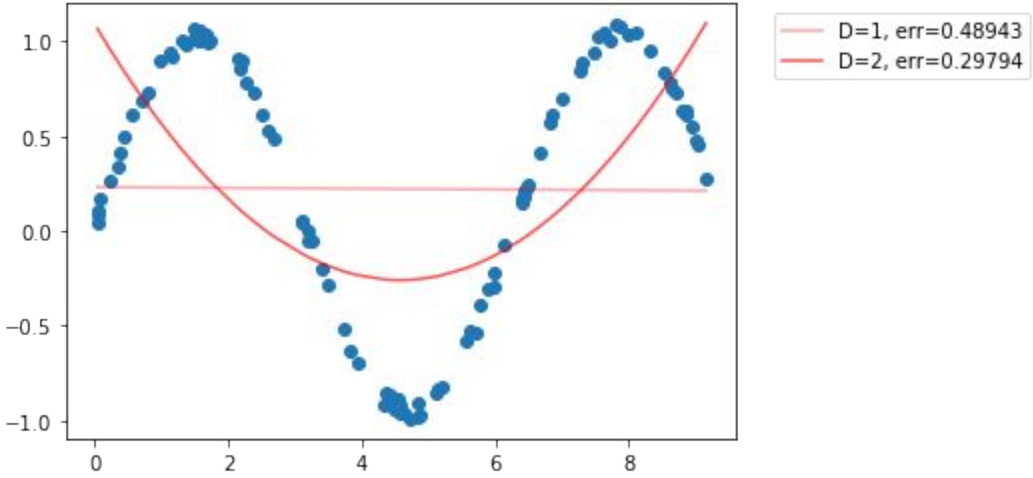
(content by Alvin Wan, John DeNero, Joseph E. Gonzalez, Josh Hug)



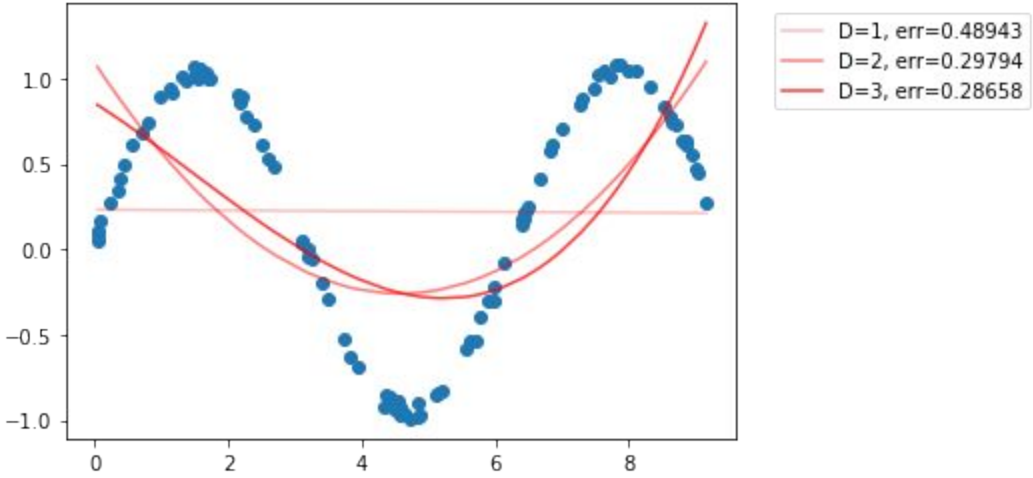
1.0
0.5
0.0
−0.5
−1.0
0
2
4
6
8

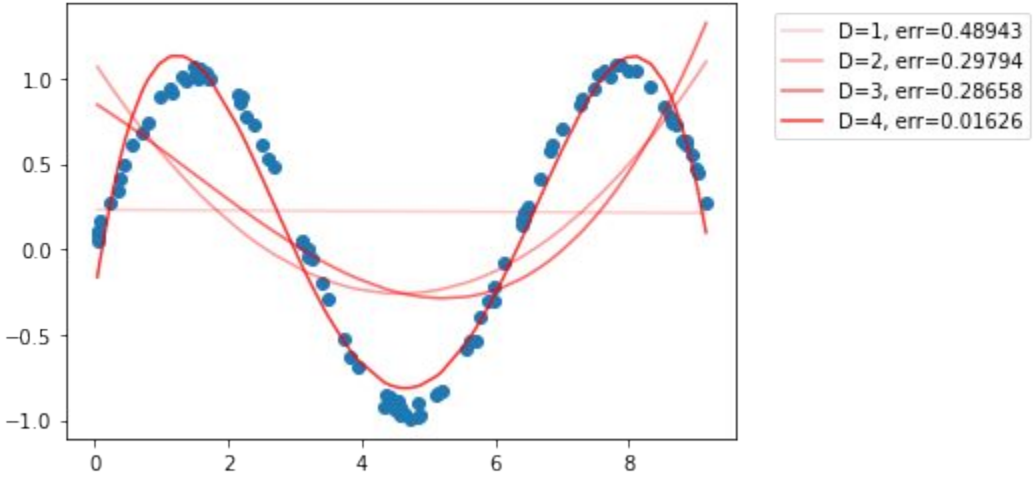


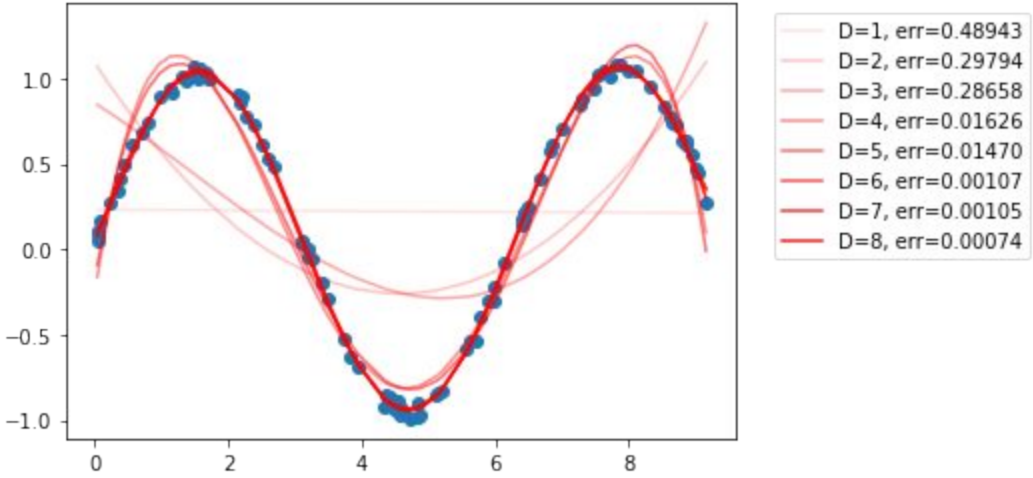
1.0
0.5
0.0
−0.5
−1.0
0
2
4
6
8
D=1, err=0.48943



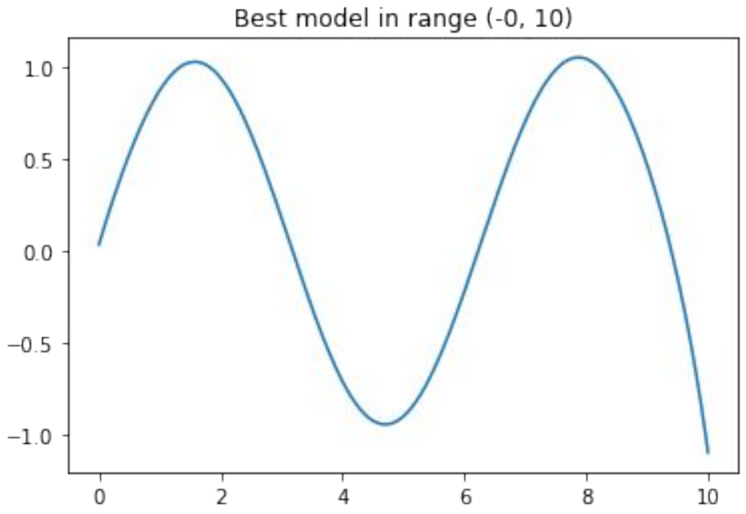
1.0
0.5
0.0
−0.5
−1.0
0
2
4
6
8
D=1, err=0.48943
D=2, err=0.29794

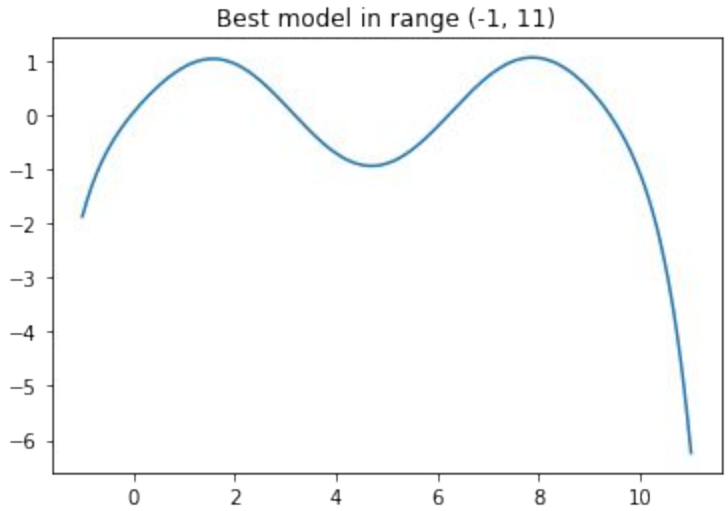

D=1, err=0.48943
D=2, err=0.29794
D=3, err=0.28658
1.0
0.5
0.0
−0.5
−1.0
0
2
4
6
8

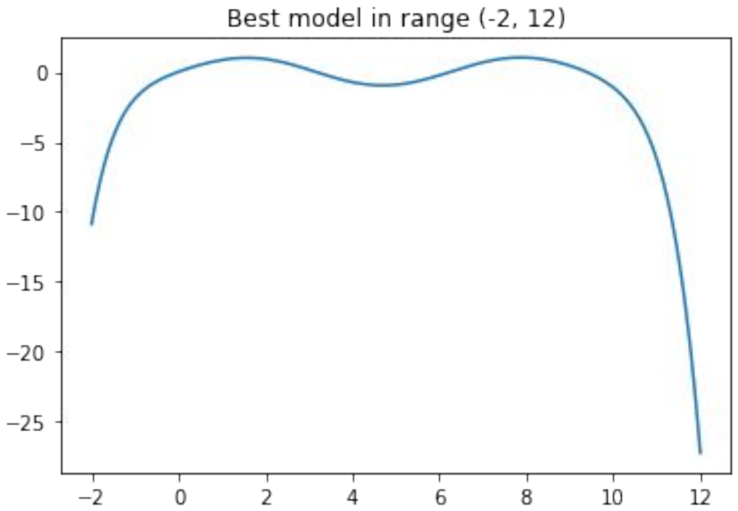

D=1, err=0.48943
D=2, err=0.29794
D=3, err=0.28658
D=4, err=0.01626
1.0
0.5
0.0
−0.5
−1.0
0
2
4
6
8

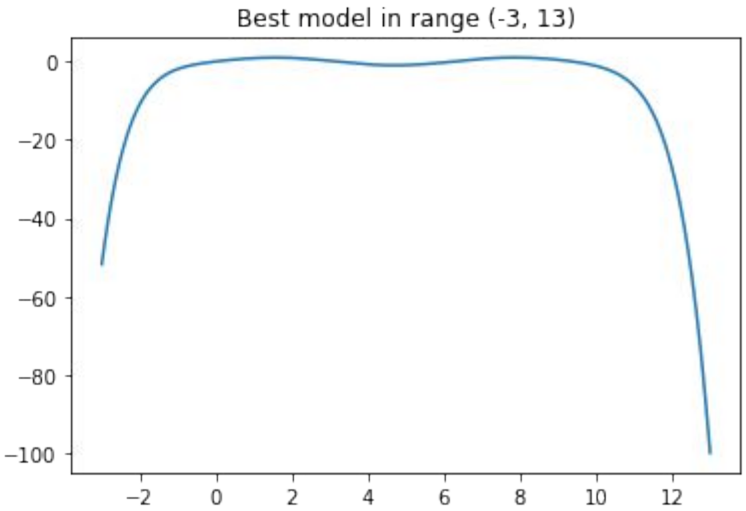


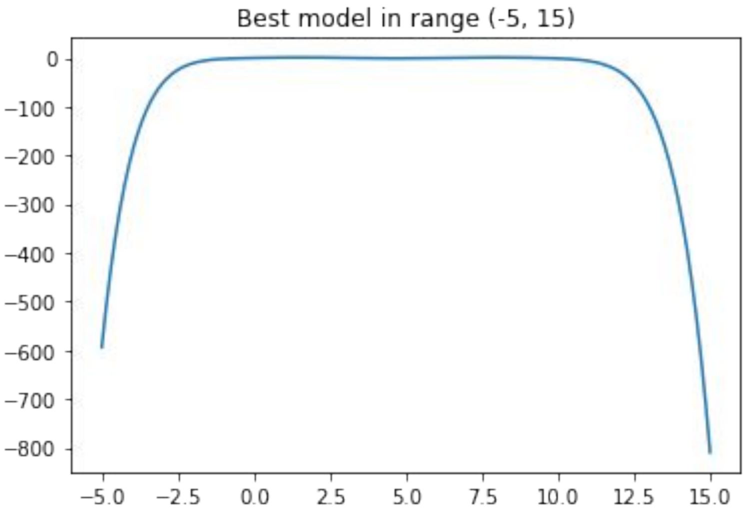
1.0
0.5
0.0
−0.5
−1.0
0
2
4
6
8
D=1, err=0.48943
D=2, err=0.29794
D=3, err=0.28658
D=4, err=0.01626
D=5, err=0.01470
D=6, err=0.00107
D=7, err=0.00105
D=8, err=0.00074


Best model in range (-0, 10)
1.0
0.5
0.0
−0.5
−1.0
0
2
4
6
8
10


Best model in range (-1, 11)
1
0
−1
−2
−3
−4
−5
−6
0
2
4
6
8
10


Best model in range (-2, 12)
0
−5
−10
−15
−20
−25
−2
0
2
4
6
8
10
12


Best model in range (-3, 13)
0
−20
−40
−60
−80
−100
−2
0
2
4
6
8
10
12


Best model in range (-5, 15)
0
−100
−200
−300
−400
−500
−600
−700
−800
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0

TAKEAWAY

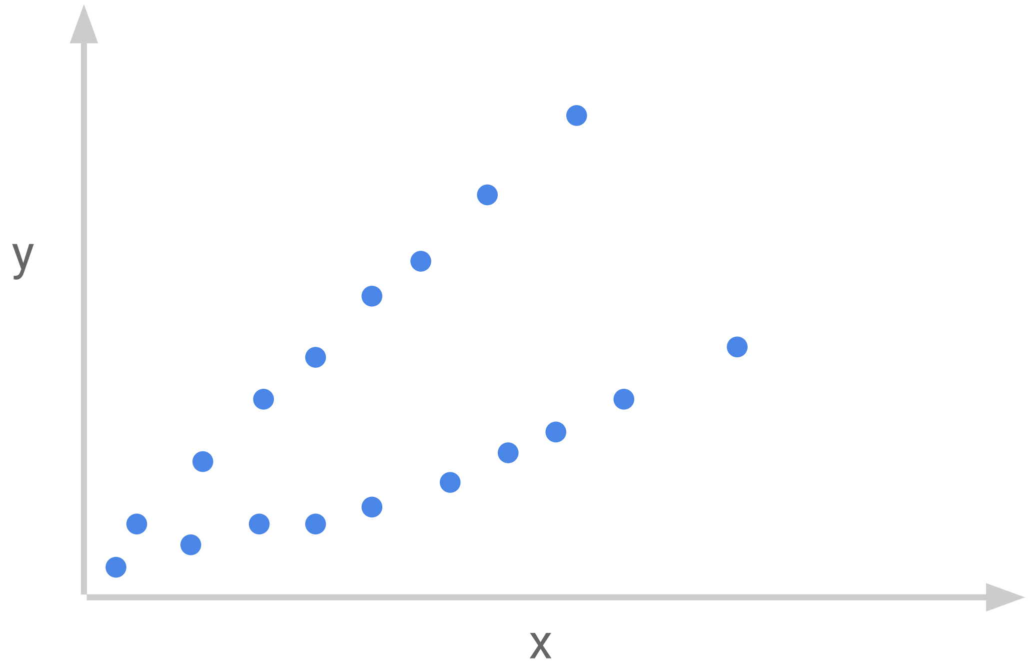
Overfitting occurs when your model **memorizes noise**, leading to **poor generalization** beyond training data.

Next Up

Regularization
Bias-Variance Tradeoff

Domain Knowledge

poor fit on (x, y)



Domain Knowledge

great fit on $(x^{t+1}-x^{t}, y)$