YaleNUSCollege

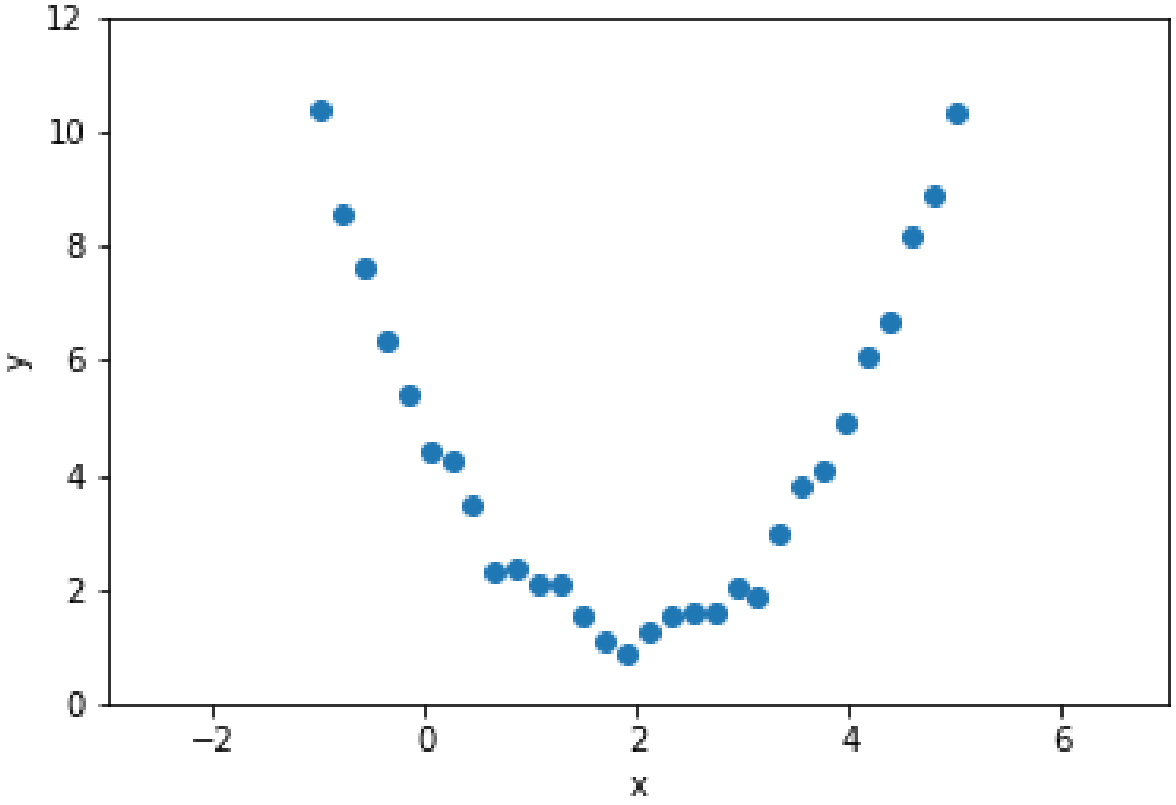
YSC2239 Lecture 16

Today’s class

- Introduction to scikit-learn
- Feature engineering

Motivating Feature Engineering

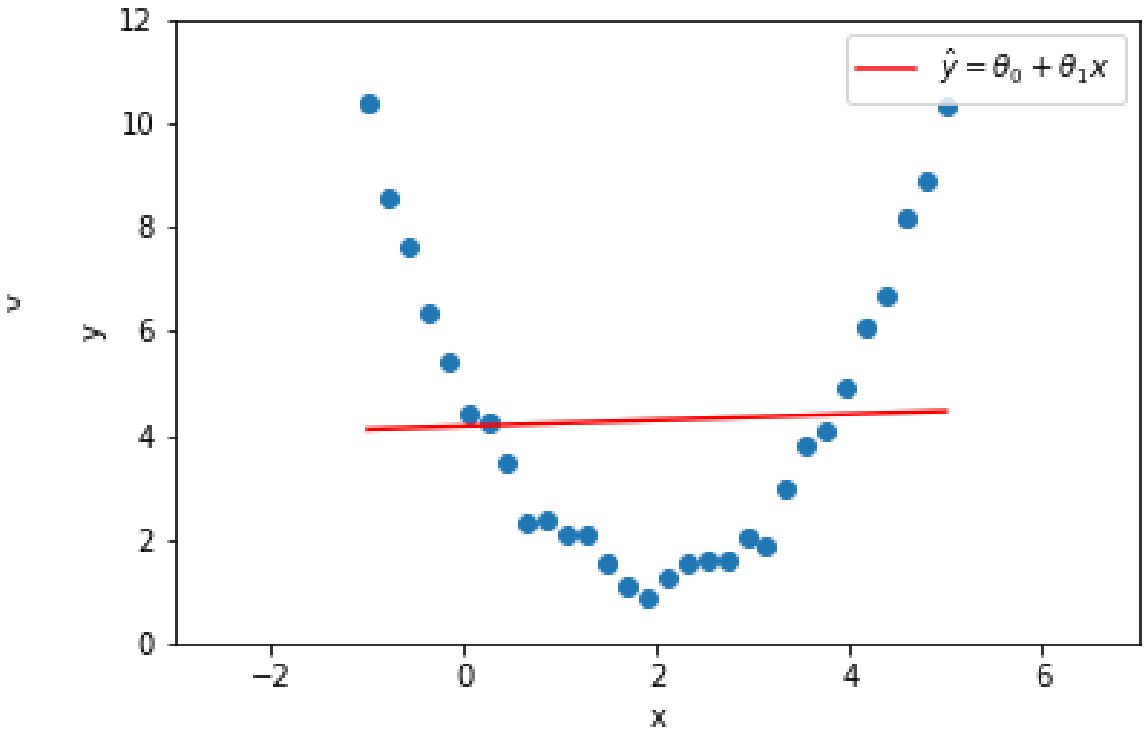
Can we fit a model to this data?



Can we fit a model to this data?

Simple Linear Regression?

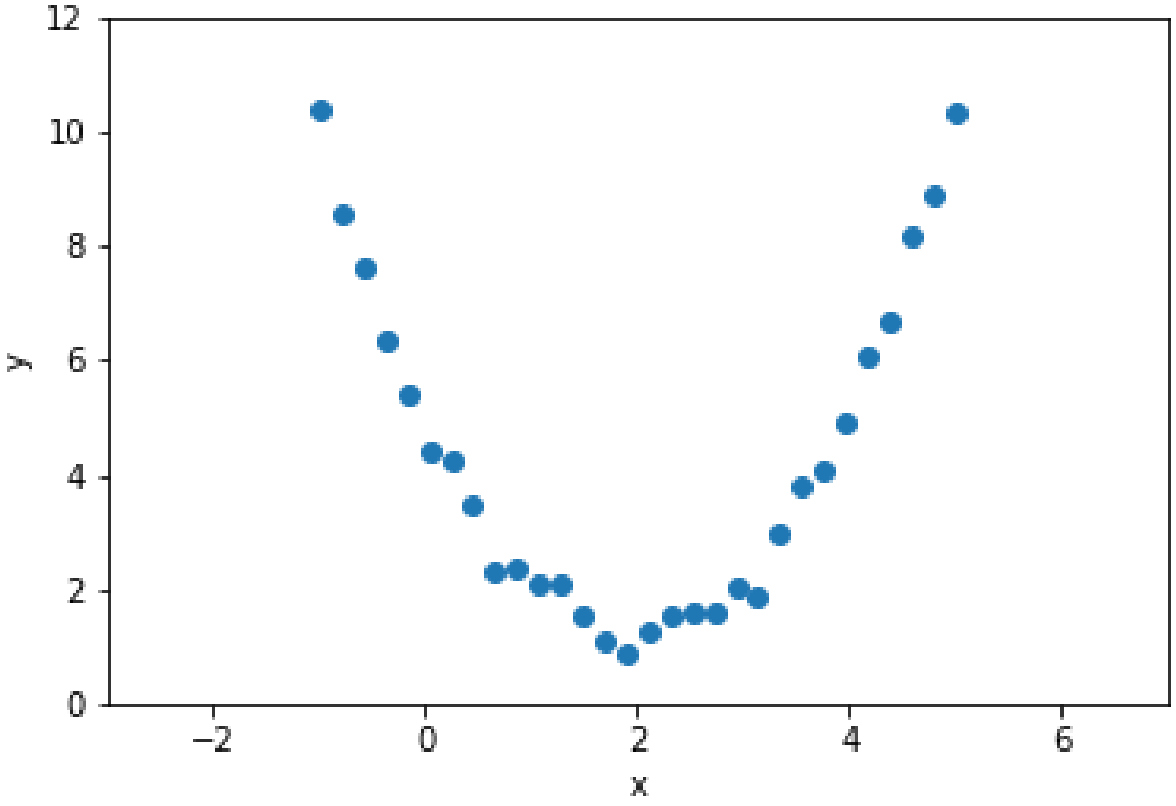
No, because the data is fundamentally nonlinear



Can we fit a model to this data?

Multiple Linear Regression?

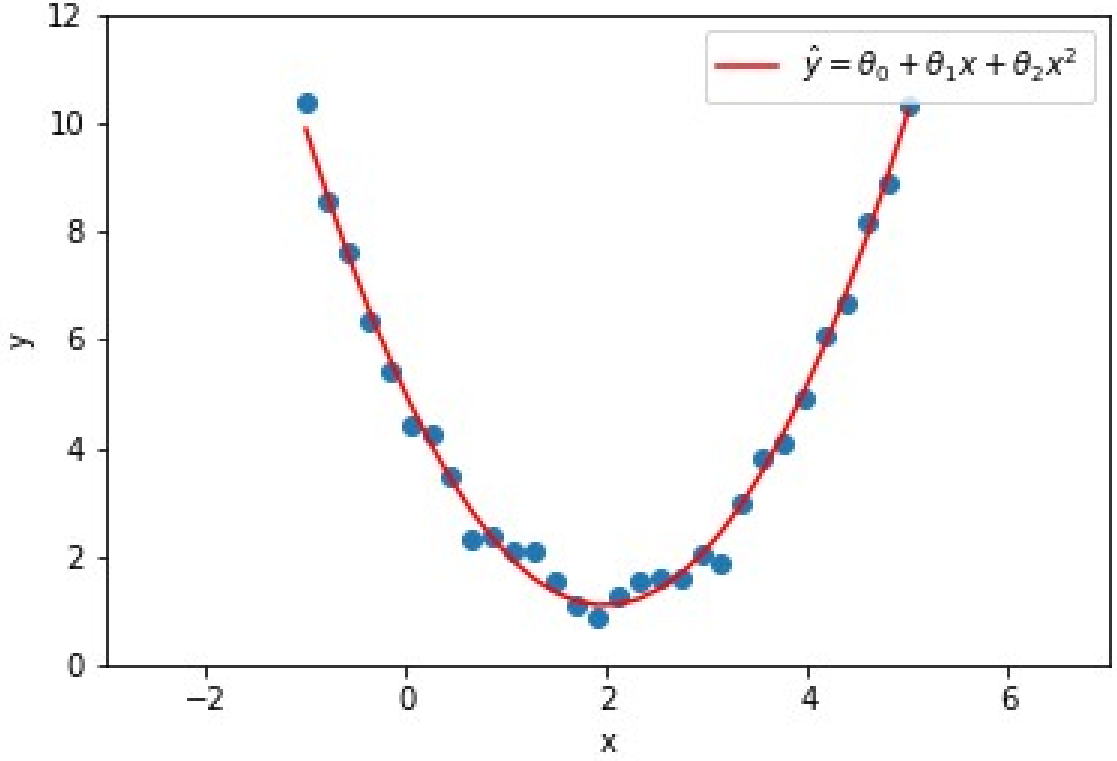
No, because there are no other features to add



Can we fit a model to this data?

Idea: *Create* an extra feature to use in the model. What feature should we add?

Since the data looks like a parabola, let's add a quadratic feature



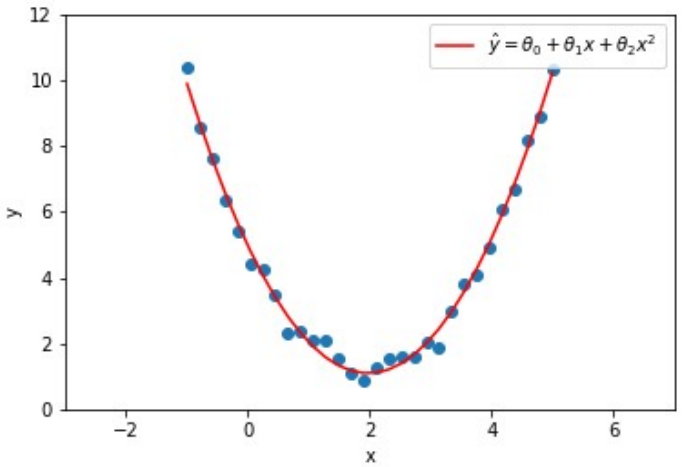
What is a feature?

A feature is an input to our model

- So far, we have just used the raw data as features

We can also create new features to use an inputs to our model

- The process of creating new features is called **feature engineering**



model

$$\theta_0 + \theta_1 x + \theta_2 x^2$$

Features: x, x^2

Parameters: θ_0, θ_1, θ_2

Can we really just create our own features?

Yes! (with some restrictions)

We can create any feature we want as long we can write the model in the form

$$\boxed{\hat{y} = x^T\theta}$$

This is a **linear combination of the features**. The features <u>*cannot*</u> depend on the parameters of the model!

Exam

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2 = \begin{bmatrix} 1 \\ x_1 \\ x_1^2 \end{bmatrix} \cdot \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix} = x^T\theta$$

Lecture Roadmap

We can choose/create **x** any way we like as long as our model follows the form $\boxed{\hat{y} = x^T\theta}$

The rest of the lecture will discuss different techniques we can use to create **x**:

- How can we create features from **quantitative data**?
- How can we create features from **categorical data**?
- How can we create features from **text data**?

Intro to Scikit-Learn (Demo)

Feature Engineering: Quantitative Data

Feature Engineering: Categorical Data

Feature Engineering: Imputation

Feature Engineering: Text Data

Bag-of-words Encoding

- Generalization of one-hot-encoding for a string of text:

"Learning about machine learning is fun."

	aardvark	aardwolf		**fun**		**learning**		**machine**		zyzzyva
Vector	0	0	…	1	…	2	…	1	…	0

- Encode text as a long vector of word counts (Issues?)
  - Typically high dimensional (millions of columns) and very sparse
  - Word order information is lost… (is this an issue?)
  - What happens when you see a word not in the dictionary?

- A **bag** is another term for a multiset: *an unordered collection which may contain multiple instances of each element.*

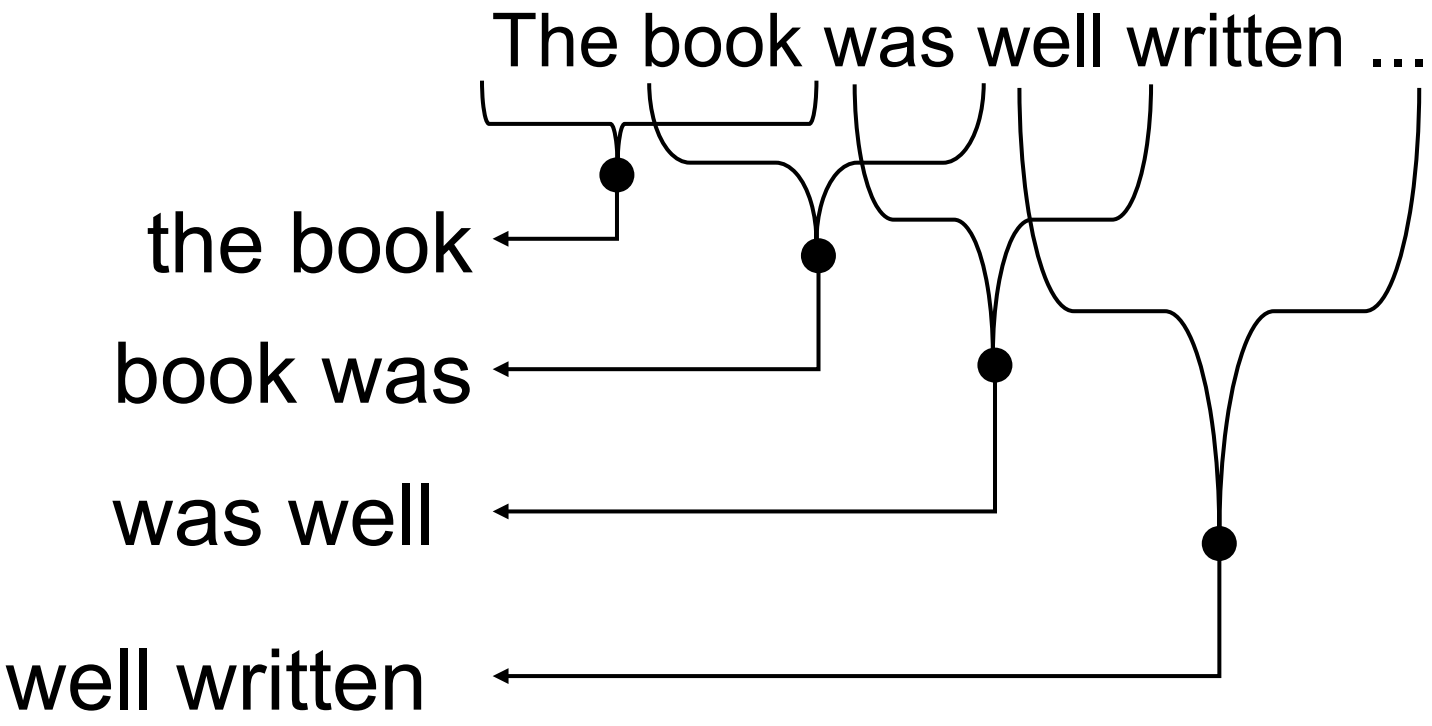
- **Stop words:** words that do not contain significant information
  - Examples: the, in, at, or, on, a, an, and …
  - Typically removed

N-Gram Encoding

- Sometimes word order matters:

*The book was **not** well written but I did enjoy it.* ➡ *The book was well written but I did **not** enjoy it.*

- How do we capture word order in a “vector” model?
  - N-Gram: *“Bag-of- sequences-of-words”*



2-Gram Encoding

	aardvark is	apple shines		**the book**		**book was**		**was well**		**well written**		zebra ran
Vector	0	0	…	1	…	1	…	1	…	1	…	0

N-Gram Encoding

- Sometimes word order matters:

*The book was **not** well written but I did enjoy it.* ➡ *The book was well written but I did **not** enjoy it.*

- How do we capture word order in a “vector” model?
  - N-Gram: *“Bag-of- sequences-of-words”*

- Issue:
  - Can be very sparse (many combinations occur only once)

Mathematical Implications of Feature Engineering

One-Hot Encoding and Linear Dependence

bias	...	co
1	...	Apple
1	...	Samsung
1	...	Apple

bias	...	co_appl	co_sam
1	...	1	0
1	...	0	1
1	...	1	0

Notice that `co_appl + co_sam = bias`! This means the columns are linearly dependent

- **Solution:** Drop one of the one-hot encoded columns per variable

Too Many Features

If you add too many features, the normal equations will have infinite solutions

The normal equations can be thought of as a system of equations with N equations and P unknown quantities to solve for

- N: # of data points
- P: # of parameters

If P > N, you have more unknowns than equations so there can be no unique solution

Additionally, too many features can cause overfitting, which will be covered in future lectures

Summary

Feature Engineering Summary

- Feature engineering is the process of creating new useful features from your data to build more sophisticated models
- Feature engineering allows you to utilize non-numerical data
  - One-hot encoding is a widely used technique
- Need to be careful in choosing how many and which features to create
  - Linearly dependent features
  - Too many features
- Feature engineering is as much an art as it is a skill
  - Neural networks try to automatically do feature engineering