Python Data Products

Course 2: Design thinking and predictive pipelines

Lecture: Features from temporal data

Learning objectives

In this lecture we will...

- Investigate different strategies for extracting features from temporal (or seasonal) data
- Extend the concept of one-hot-encodings to represent temporal information

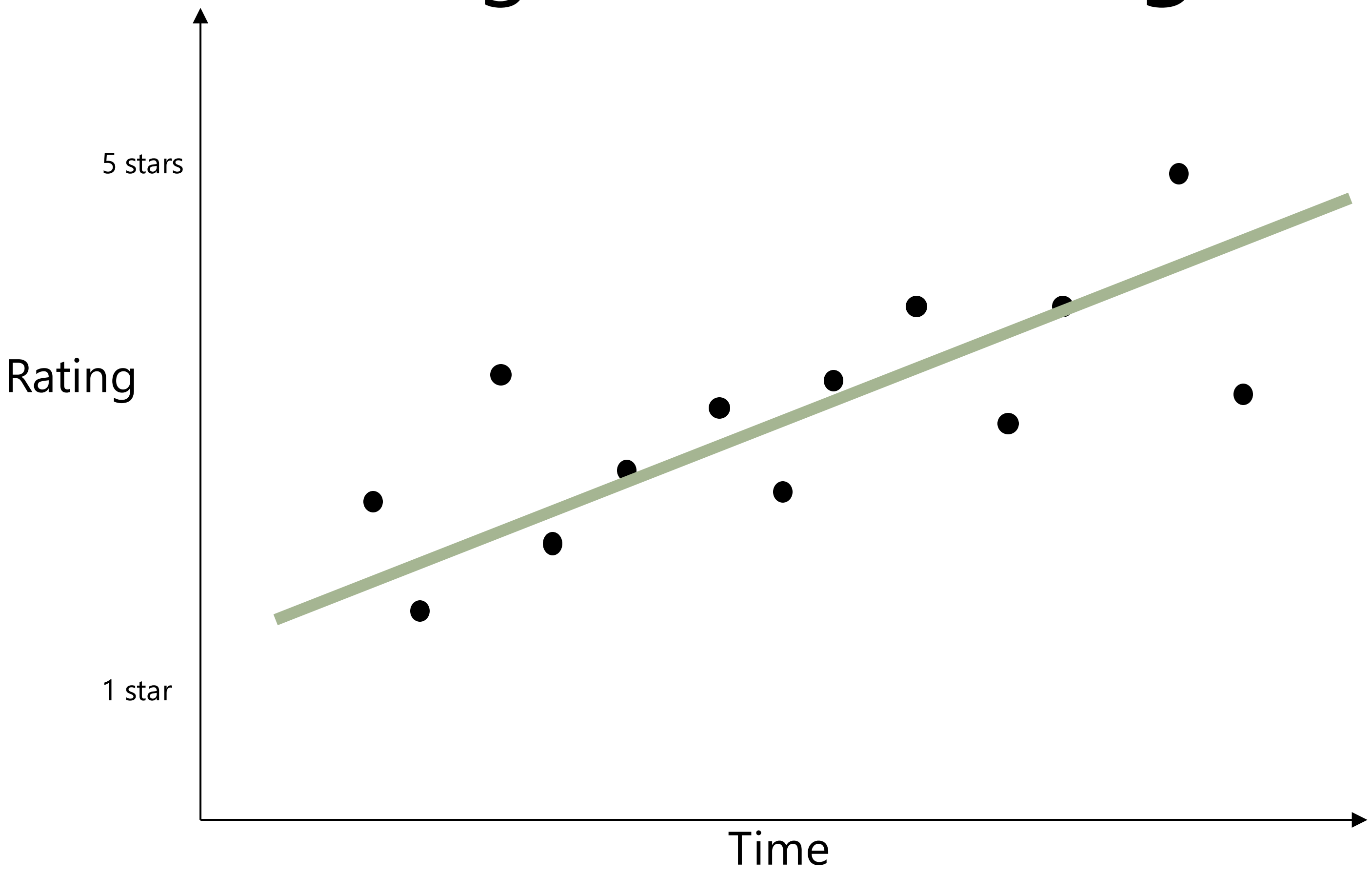
Motivating examples

How would we build regression models that incorporate features like:

- How do sales (or preferences) vary over time?
- What are the **long term** trends of sales?
- What are the **short term** trends (e.g. day of the week, season, etc.)

Motivating examples

E.g. How do **ratings** vary with **time**?



Motivating examples

E.g. How do **ratings** vary with **time**?

- In principle this picture looks okay (compared our previous lecture on categorical features) – we're predicting a **real valued** quantity from **real valued** data (assuming we convert the date string to a number)
- So, what would happen if (e.g. we tried to train a predictor based on the month of the year)?

Motivating examples

E.g. How do **ratings** vary with **time**?

- Let's start with a simple feature representation, e.g. map the month name to a month number:

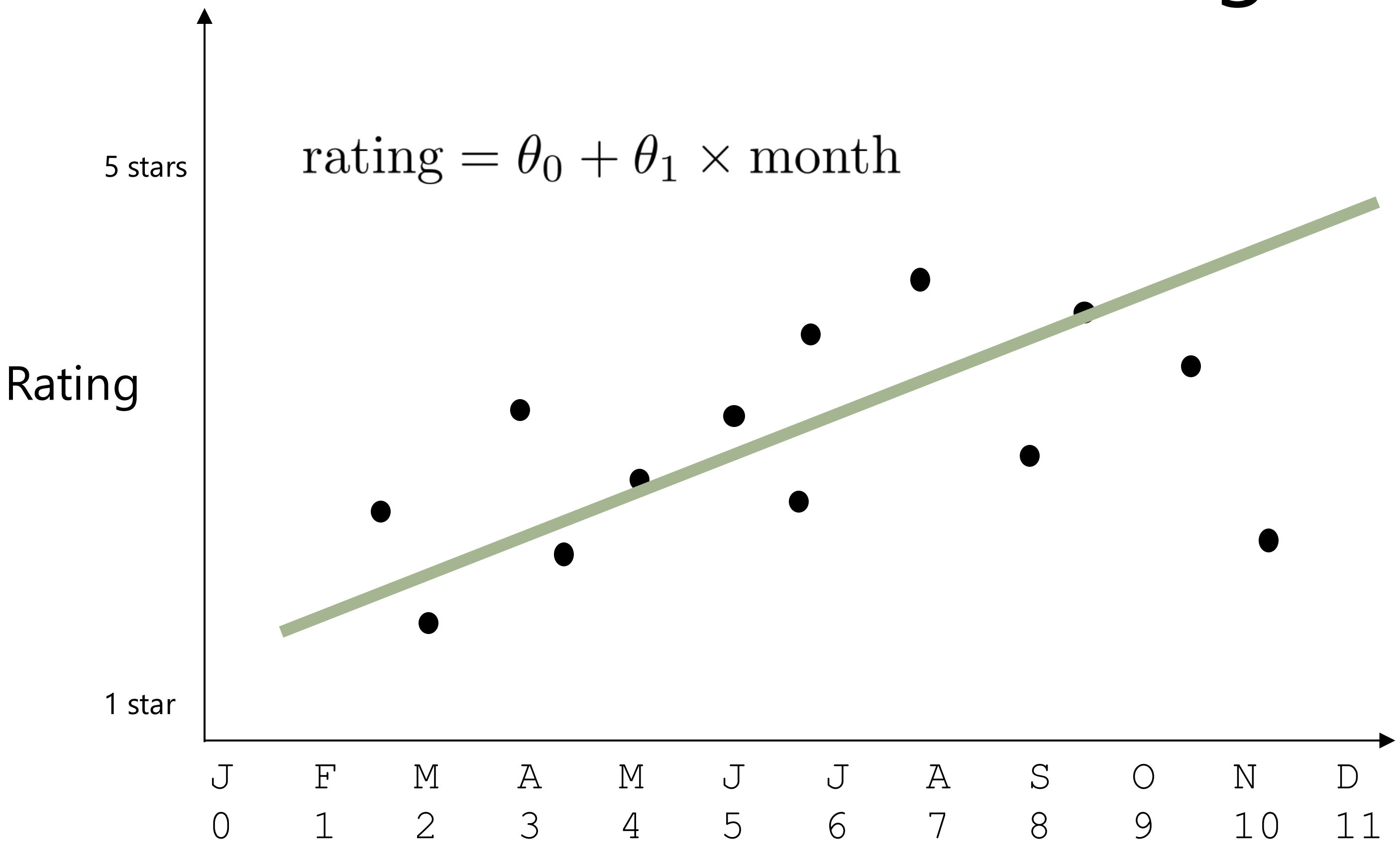
$$\text{rating} = \theta_0 + \theta_1 \times \text{month}$$

where

```
Jan = [0]
Feb = [1]
Mar = [2]
etc.
```

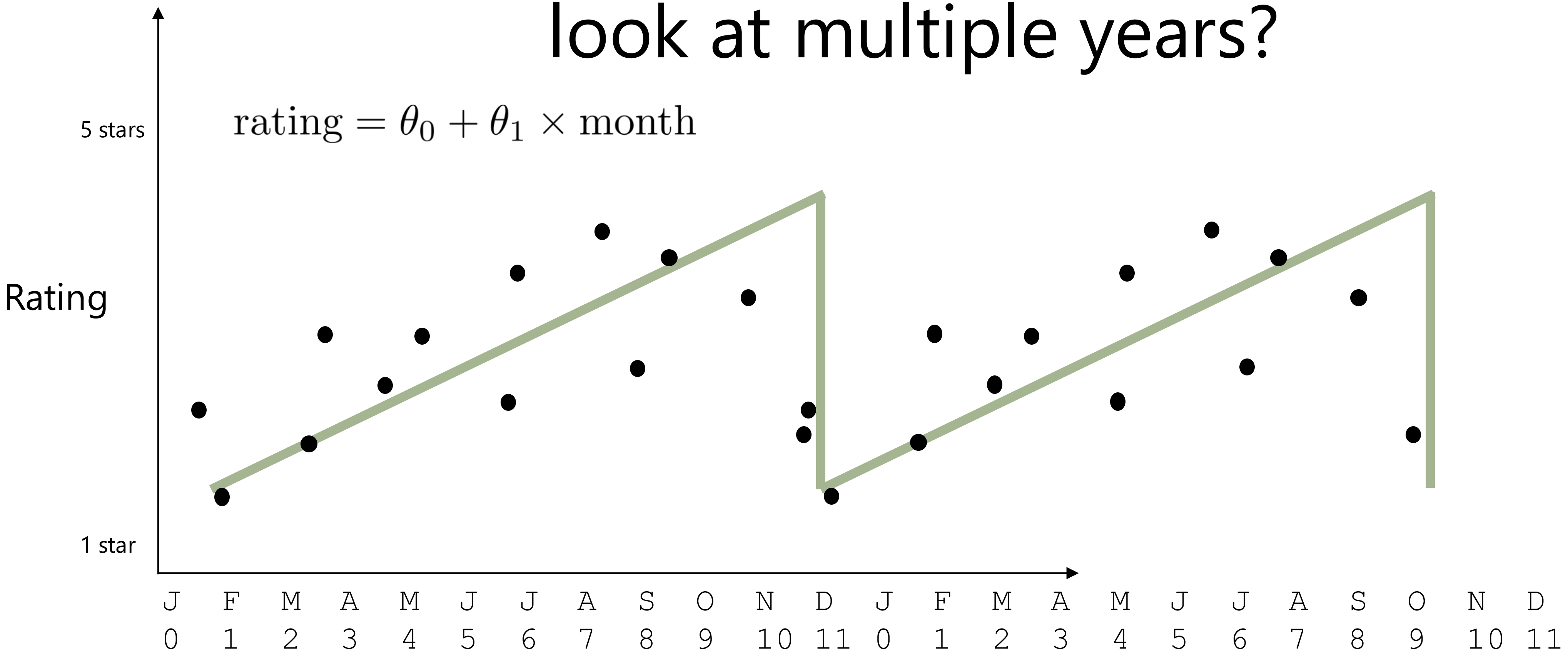
Motivating examples

The model we'd learn might look something like:



Motivating examples

This seems fine, but what happens if we look at multiple years?

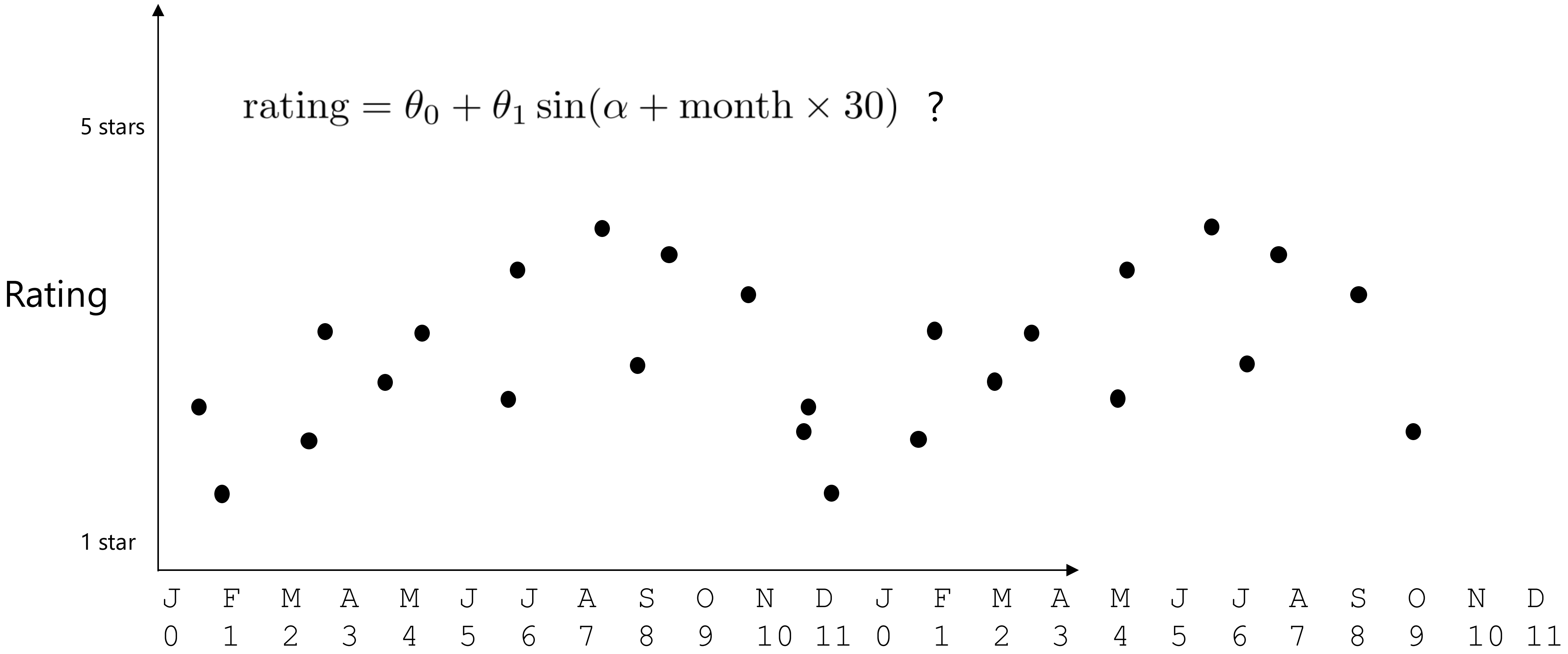


This seems fine, but what happens if we look at multiple years?

- This representation implies that the model would "wrap around" on December 31 to its January 1st value.
- This type of "sawtooth" pattern probably isn't very realistic

Modeling temporal data

What might be a more realistic shape?



Modeling temporal data

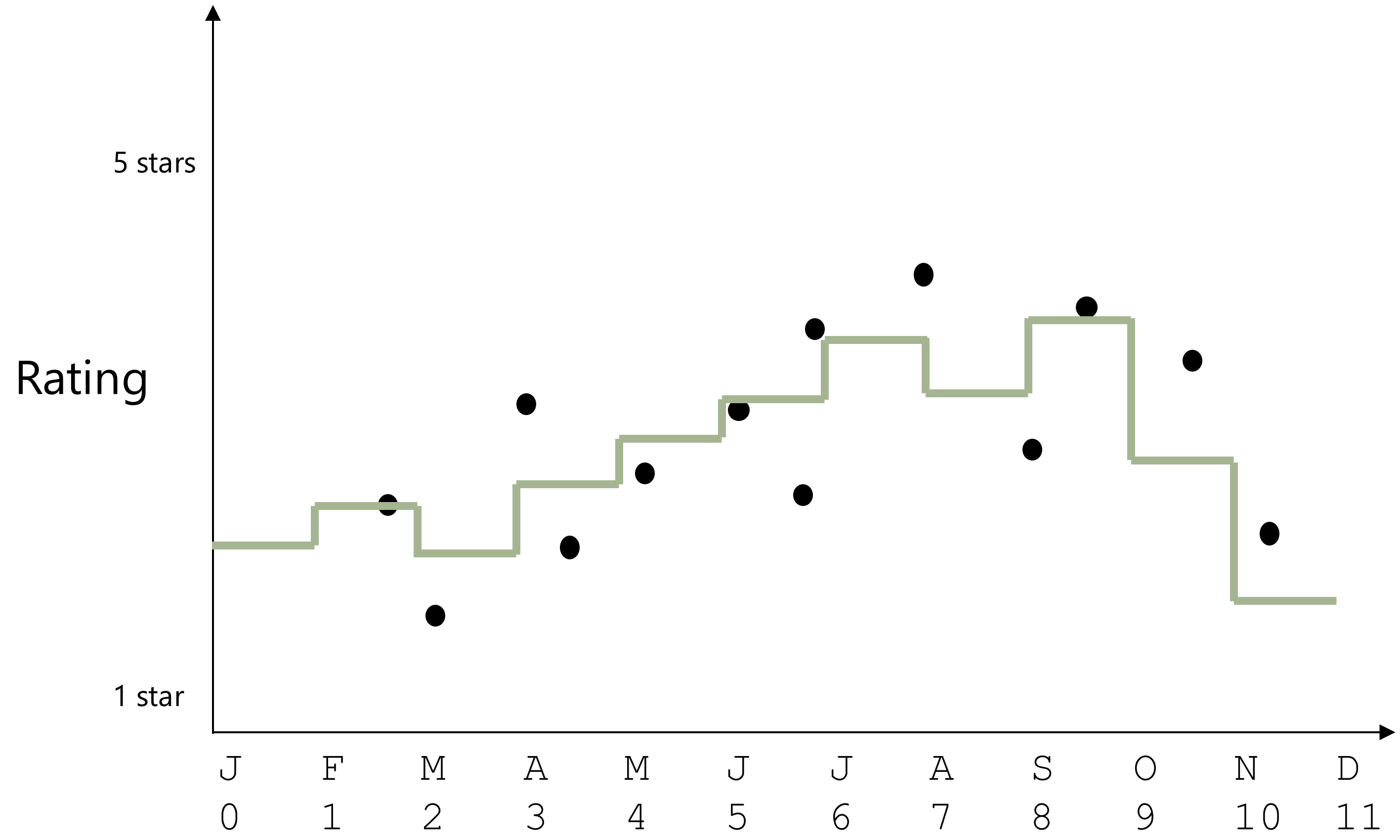
Fitting some periodic function like a sin wave would be a valid solution, but is difficult to get right, and fairly inflexible

- Also, it's not a **linear model**

- **Q:** What's a class of functions that we can use to capture a more flexible variety of shapes?
- **A:** Piecewise functions!

Concept: Fitting piecewise functions

We'd like to fit a function like the following:



Fitting piecewise functions

In fact this is very easy, even for a linear model! This function looks like:

$$\text{rating} = \theta_0 + \theta_1 \times \delta(\text{is Feb}) + \theta_2 \times \delta(\text{is Mar}) + \theta_3 \times \delta(\text{is Apr}) \cdots$$

1 if it's Feb, 0 otherwise

- Note that we don't need a feature for January
- i.e., theta_0 captures the January value, theta_0 captures the *difference* between February and January, etc.

Fitting piecewise functions

Or equivalently we'd have features as follows:

$$\text{rating} = \theta \cdot x \quad \text{where}$$

```
x = [1,1,0,0,0,0,0,0,0,0,0,0] if February
    [1,0,1,0,0,0,0,0,0,0,0,0] if March
    [1,0,0,1,0,0,0,0,0,0,0,0] if April
    ...
    [1,0,0,0,0,0,0,0,0,0,0,1] if December
```

Fitting piecewise functions

Note that this is still a form of **one-hot** encoding, just like we saw in the "categorical features" lecture

- This type of feature is very flexible, as it can handle complex shapes, periodicity, etc.
- We could easily increase (or decrease) the resolution to a week, or an entire season, rather than a month, depending on how fine-grained our data was

Concept: Combining one-hot encodings

We can also extend this by combining several one-hot encodings together:

$$\text{rating} = \theta \cdot x = \theta \cdot [x_1; x_2] \text{ where}$$

```
x1 = [1,1,0,0,0,0,0,0,0,0,0,0] if February
     [1,0,1,0,0,0,0,0,0,0,0,0] if March
     [1,0,0,1,0,0,0,0,0,0,0,0] if April
     ...
     [1,0,0,0,0,0,0,0,0,0,0,1] if December
```

```
x2 = [1,0,0,0,0,0] if Tuesday
     [0,1,0,0,0,0] if Wednesday
     [0,0,1,0,0,0] if Thursday
     ...
```

Summary of concepts

- Motivated the use of piecewise functions to model temporal data
- Described how one-hot encodings can be used to model piecewise functions

On your own...

- Think about what piecewise functions you might use to model demand on Amazon
  - Is the day of the week important?
  - Or the day of the month?
- How would you incorporate significant holidays (which may influence demand) into this model?