Polynomial Regression

Polynomial Regression

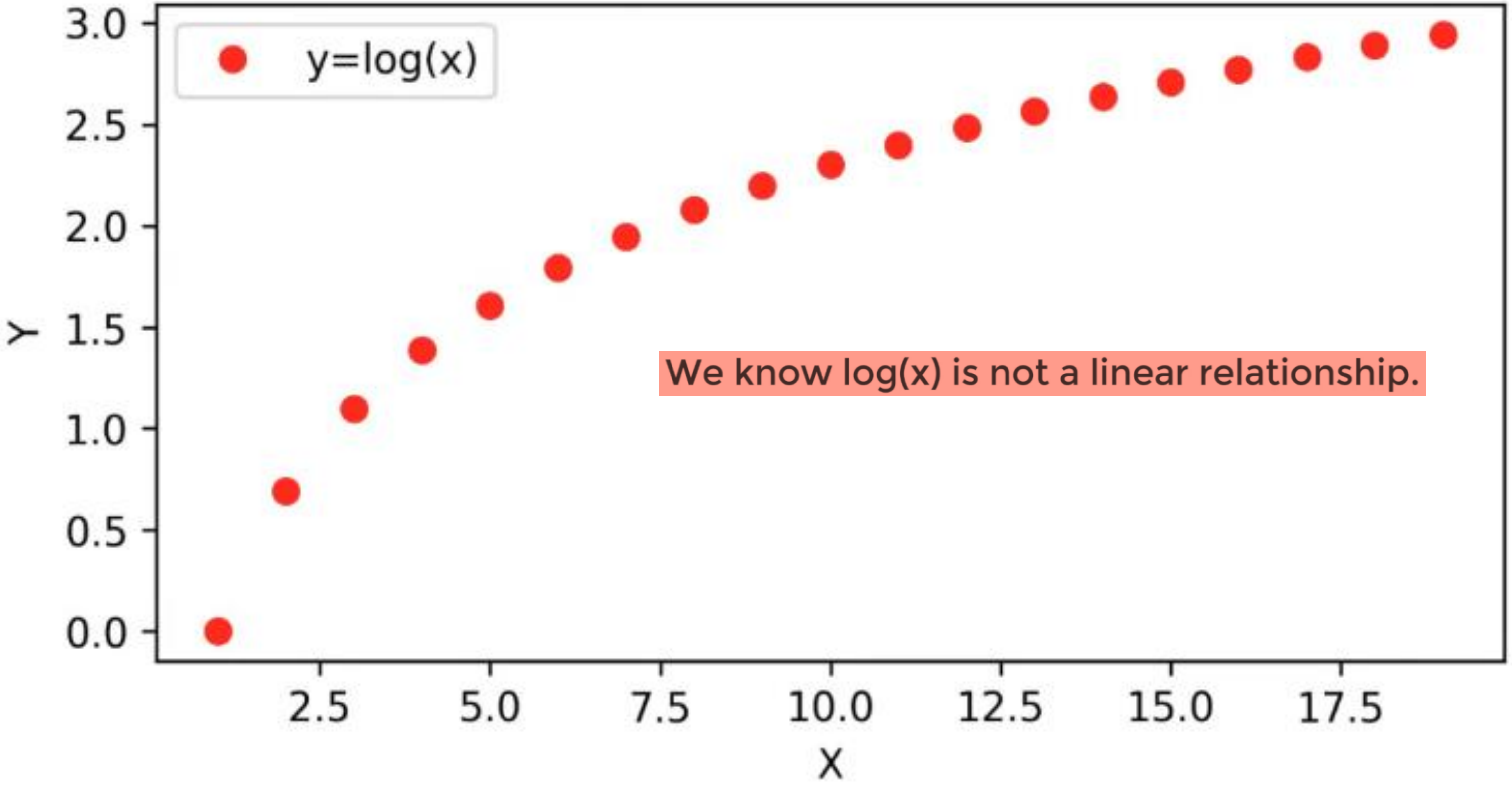
- We just completed a Linear Regression task, allowing us to predict future label values given a set of features!
- How can we now improve on a Linear Regression model?
- One approach is to consider **higher order relationships** on the features.

Polynomial Regression

- There are two issues polynomial regression will address for us:
  - Non-linear feature relationships to label
  - Interaction terms between features
- Let’s first explore non-linear relationships and how considering polynomial orders could help address this.

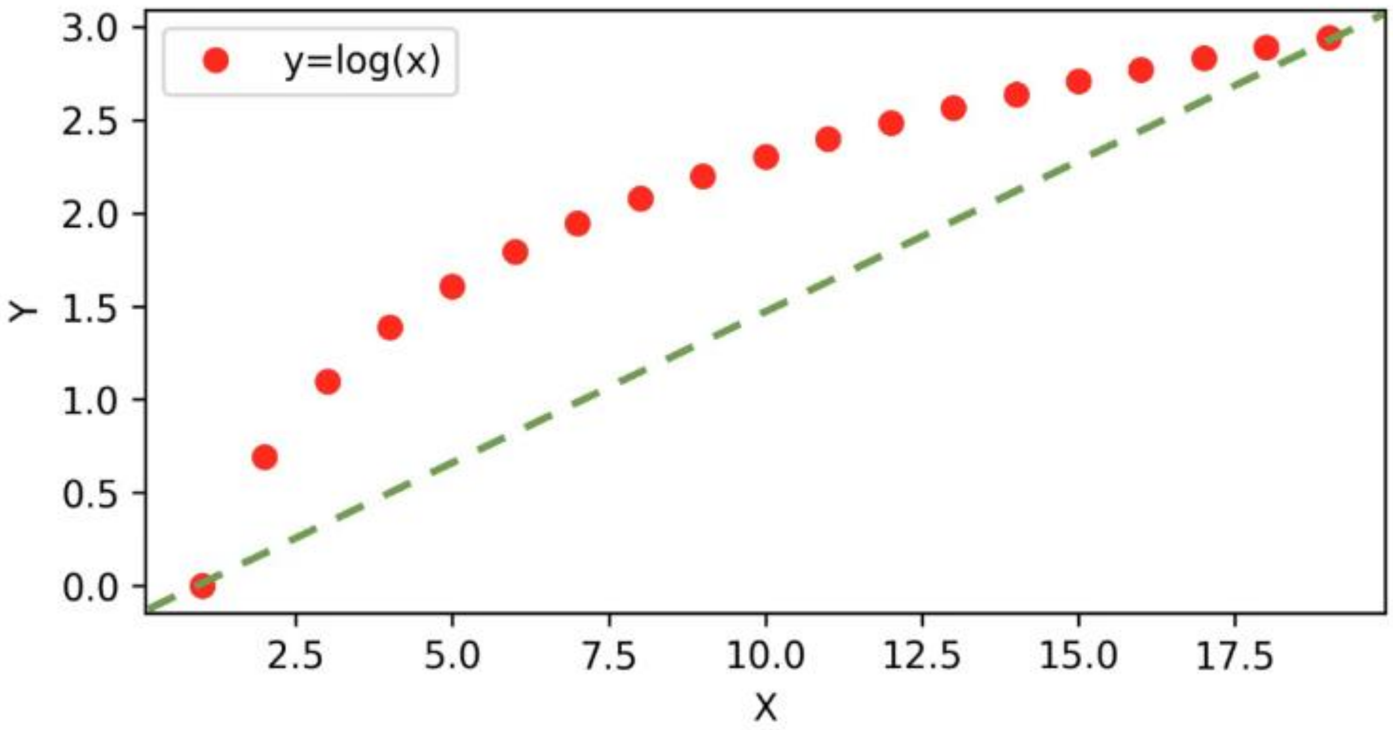
Polynomial Regression

- Imagine a feature that is not linear:



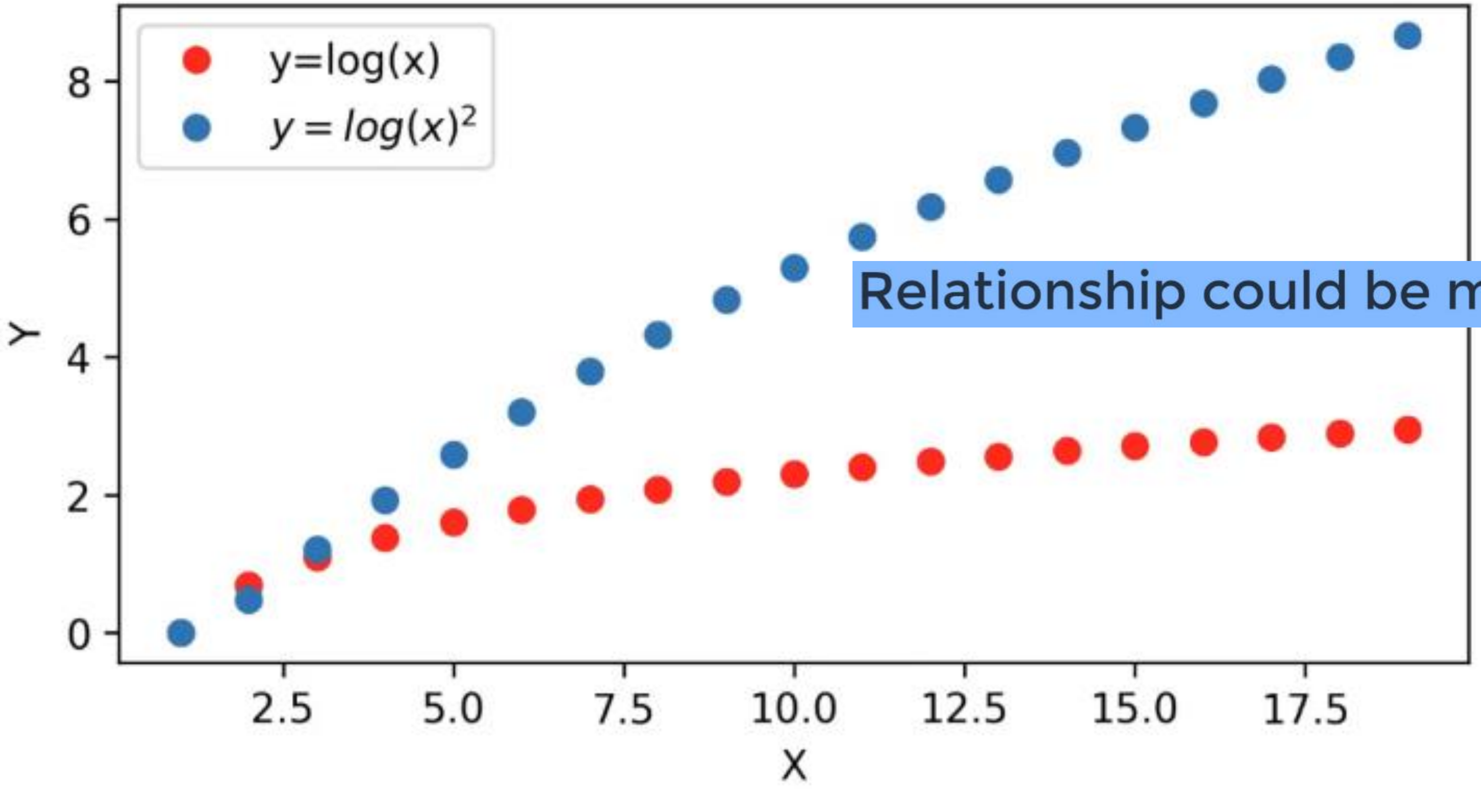
Polynomial Regression

- Will be difficult to find a linear relationship



Polynomial Regression

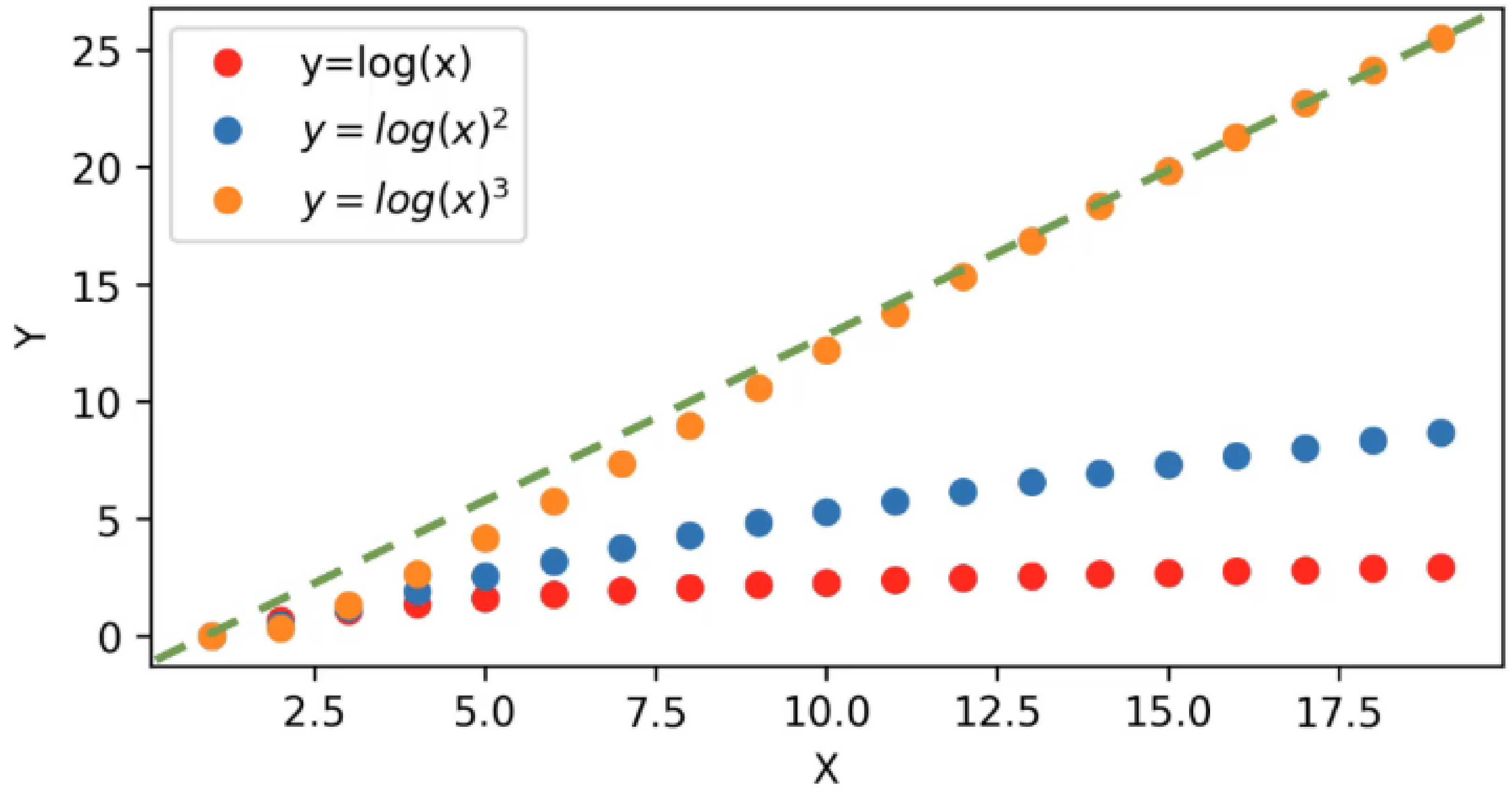
- What about the square of this feature?



Relationship could be more linear.

Polynomial Regression

- Even more so for higher orders!



Polynomial Regression

- Keep in mind this is an exaggerated example, and not every feature will have relationships at a higher order.
- The main point here is to show it could be reasonable to solve for a single linear Beta coefficient for polynomial of an original feature.

Polynomial Regression

- Let's now also consider **interaction terms.**
- What if features are only significant when in sync with one another?
- For example:
    - Perhaps newspaper advertising spend by itself is not effective, but greatly increases effectiveness if added to a TV advertising campaign.

Polynomial Regression

- Consumers only watching a TV ad will create some sales, but consumers who watch TV **and** are later “reminded” through a newspaper ad could contribute even more sales than TV or newspaper alone!
- How can we check for this?

Polynomial Regression

- Simplest way is to create a new feature that multiplies two existing features together to create an **interaction term.**
- We can keep the original features, and add on this **interaction term.**
- Fortunately Scikit-Learn does this for us easily through a **preprocessing** call.

Polynomial Regression

- Scikit-Learn's preprocessing library contains many useful tools to apply to the original data set **before** model training.
- One tool is the **PolynomialFeatures** which automatically creates both higher order feature polynomials and the interaction terms between all feature combinations.

Polynomial Regression

- The features created include:
  - The bias (the value of 1.0)
  - Values raised to a power for each degree (e.g. x^1, x^2, x^3, …)
  - Interactions between all pairs of features (e.g. x1 * x2, x1 * x3, …)

Polynomial Regression

- Converting Two Features **A** and **B**
  - **1, A, B, A^2, AB, B^2**

- Generalized terms of features X_1 and X_2
  - $1, \boxed{X_1}, \boxed{X_2}, \boxed{X_1^2}, \boxed{X_1X_2}, \boxed{X_2^2}$
- Example if row was X_1=2 and X_2=3
  - $1, \boxed{2}, \boxed{3}, \boxed{4}, \boxed{6}, \boxed{9}$

Polynomial Regression

From Preprocessing, import PolynomialFeatures, which will help us transform our original data set by adding polynomial features

We will go from the equation in the form (shown here as if we only had one x feature):

$$\hat{y} = \beta_0 + \beta_1 x_1 + \epsilon$$

and create more features from the original x feature for some *d* degree of polynomial.

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_1 x_1^2 + \ldots + \beta_d x_1^d + \epsilon$$

Then we can call the linear regression model on it, since in reality, we're just treating these new polynomial features x^2, x^3, ... x^d as new features. Obviously we need to be careful about choosing the correct value of *d* , the degree of the model. Our metric results on the test set will help us with this!

The other thing to note here is we have multiple X features, not just a single one as in the formula above, so in reality, the PolynomialFeatures will also take *interaction* terms into account for example, if an input sample is two dimensional and of the form [a, b], the degree-2 polynomial features are [1, a, b, a^2, ab, b^2].

Polynomial Regression

- Let’s explore how to perform polynomial regression with Scikit-Learn in the next lecture!