Polynomial Regression

Theory and Motivation

Polynomial Regression

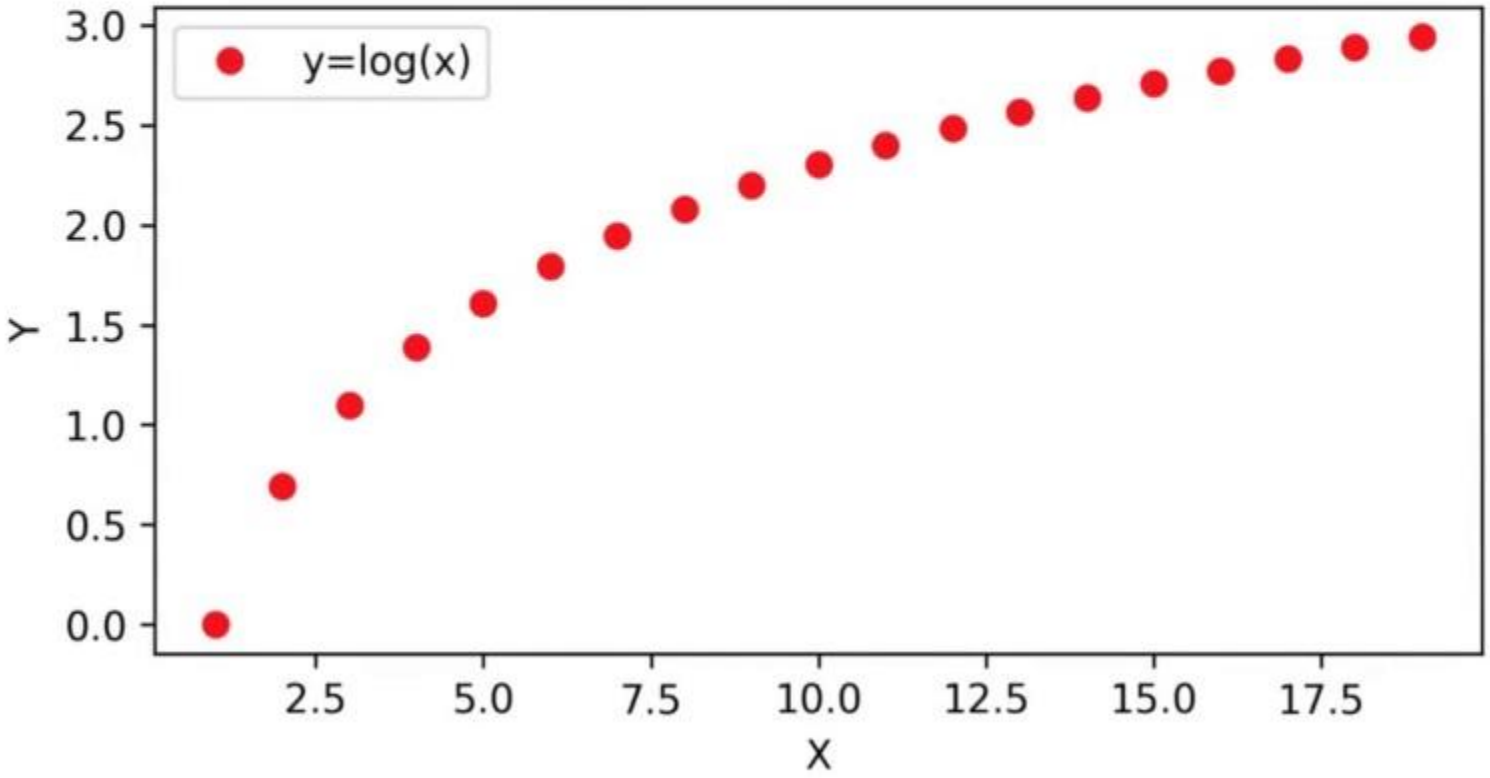
- We just completed a Linear Regression task, allowing us to predict future label values given a set of features!
- How can we now improve on a Linear Regression model?
- One approach is to consider **higher order relationships** on the features.

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.

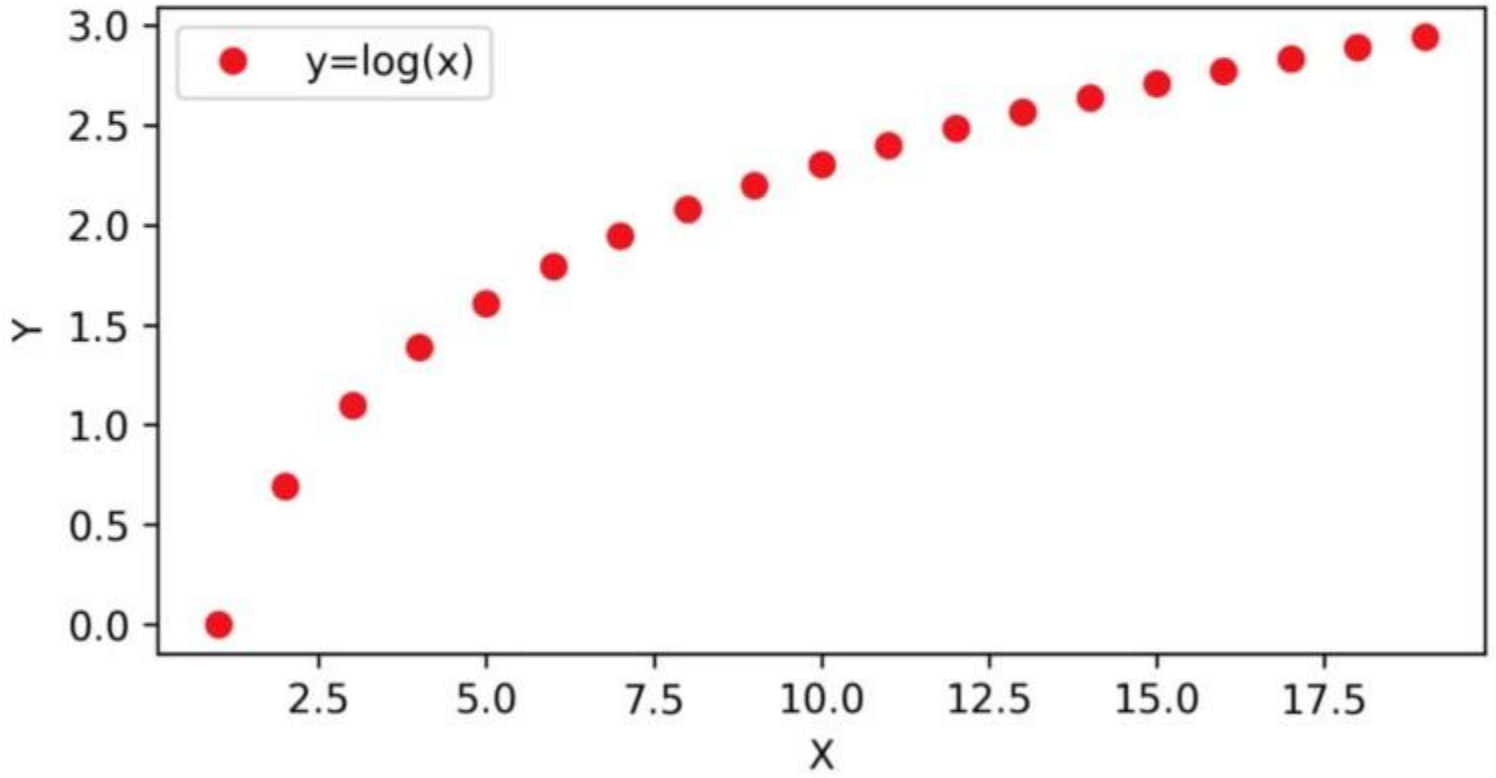
Polynomial Regression

- Imagine a feature that is not linear:



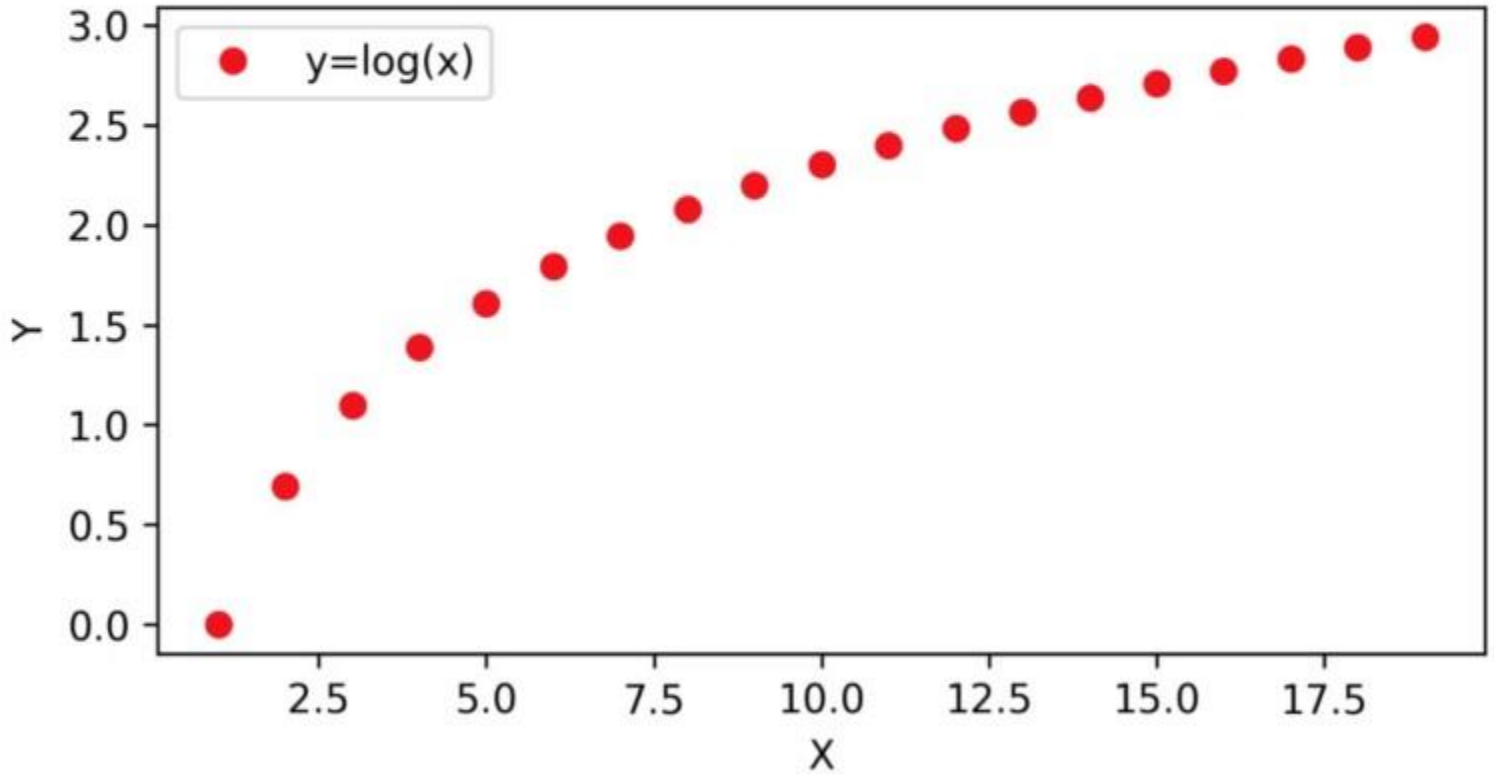
Polynomial Regression

- We know log(x) is not a linear relationship.



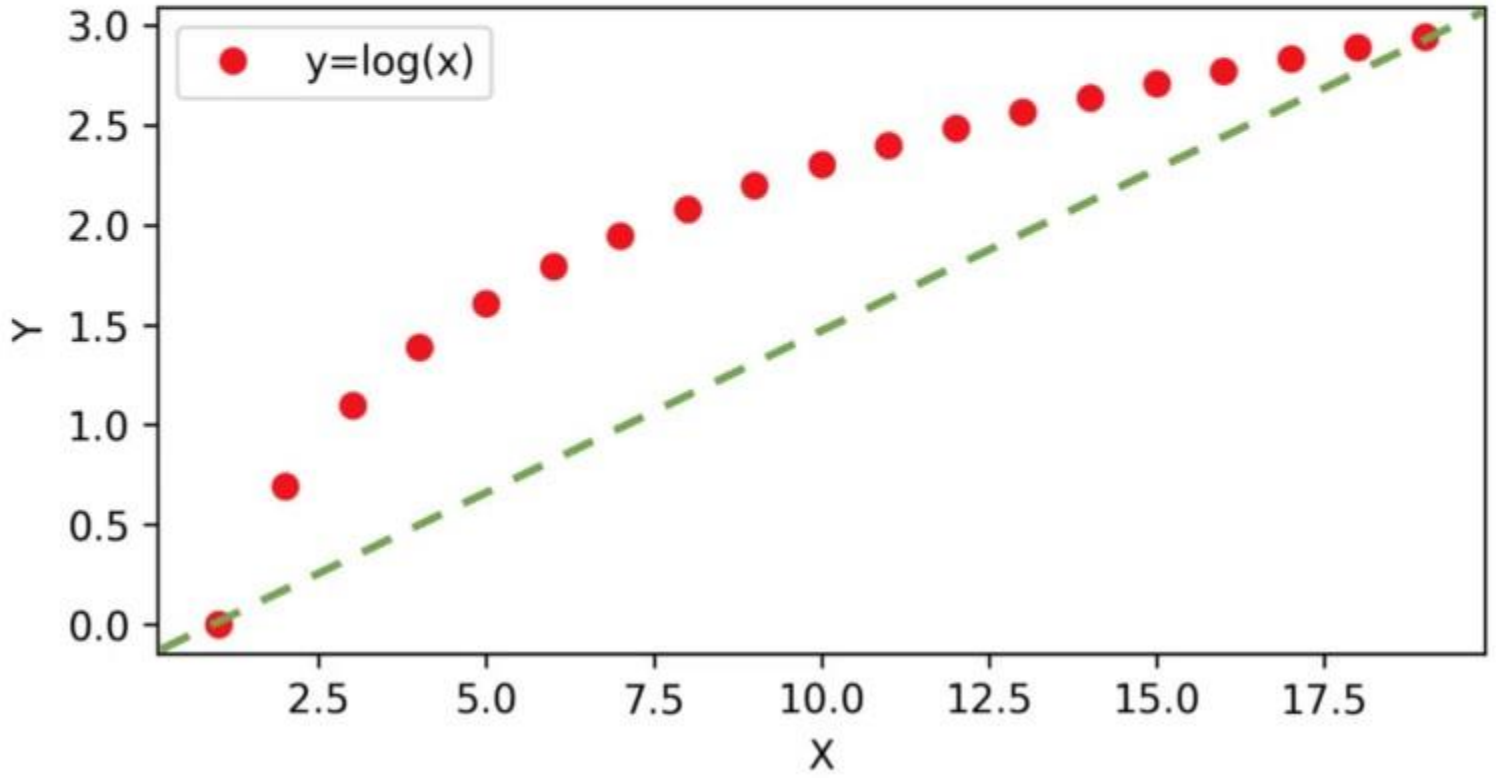
Polynomial Regression

- What is a feature X behaved like log(x)?



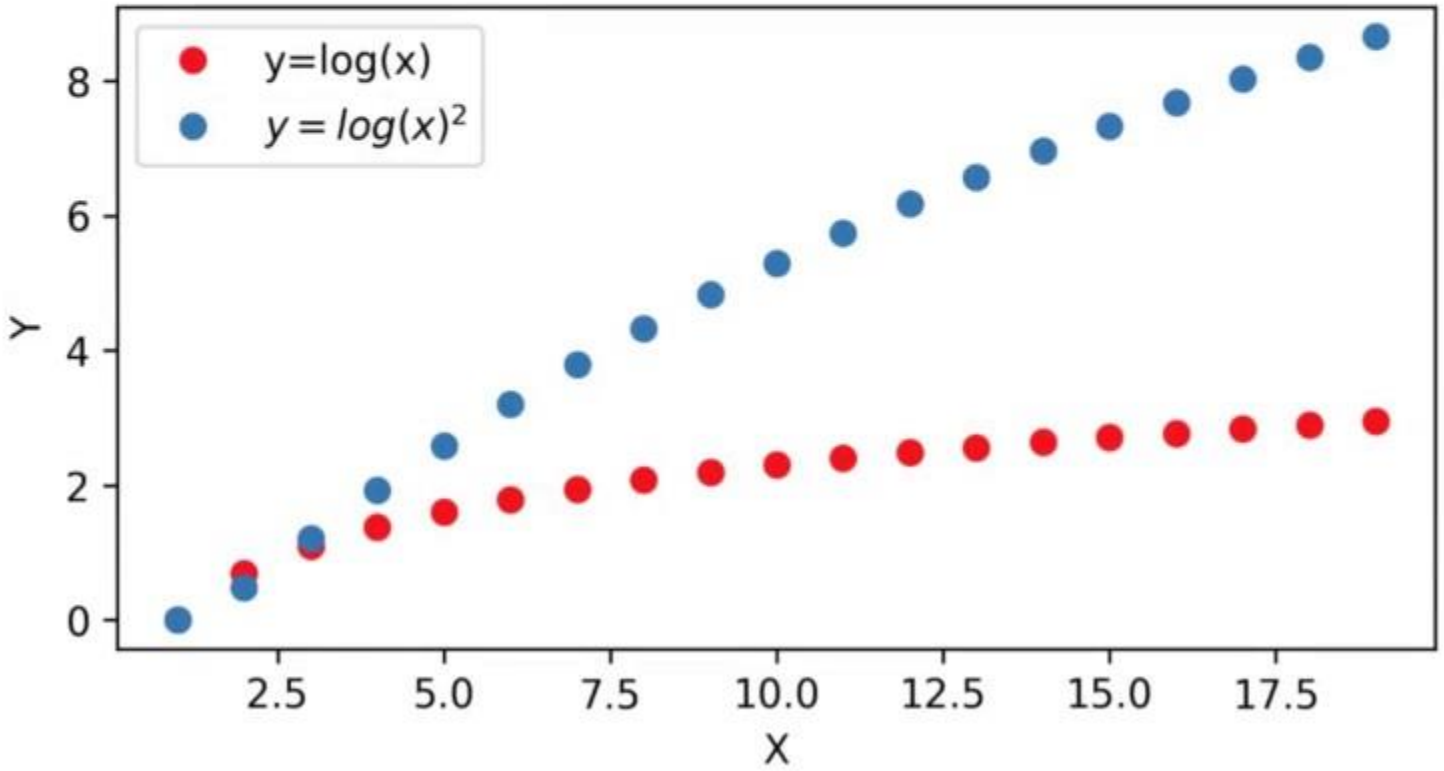
Polynomial Regression

- Will be difficult to find a linear relationship



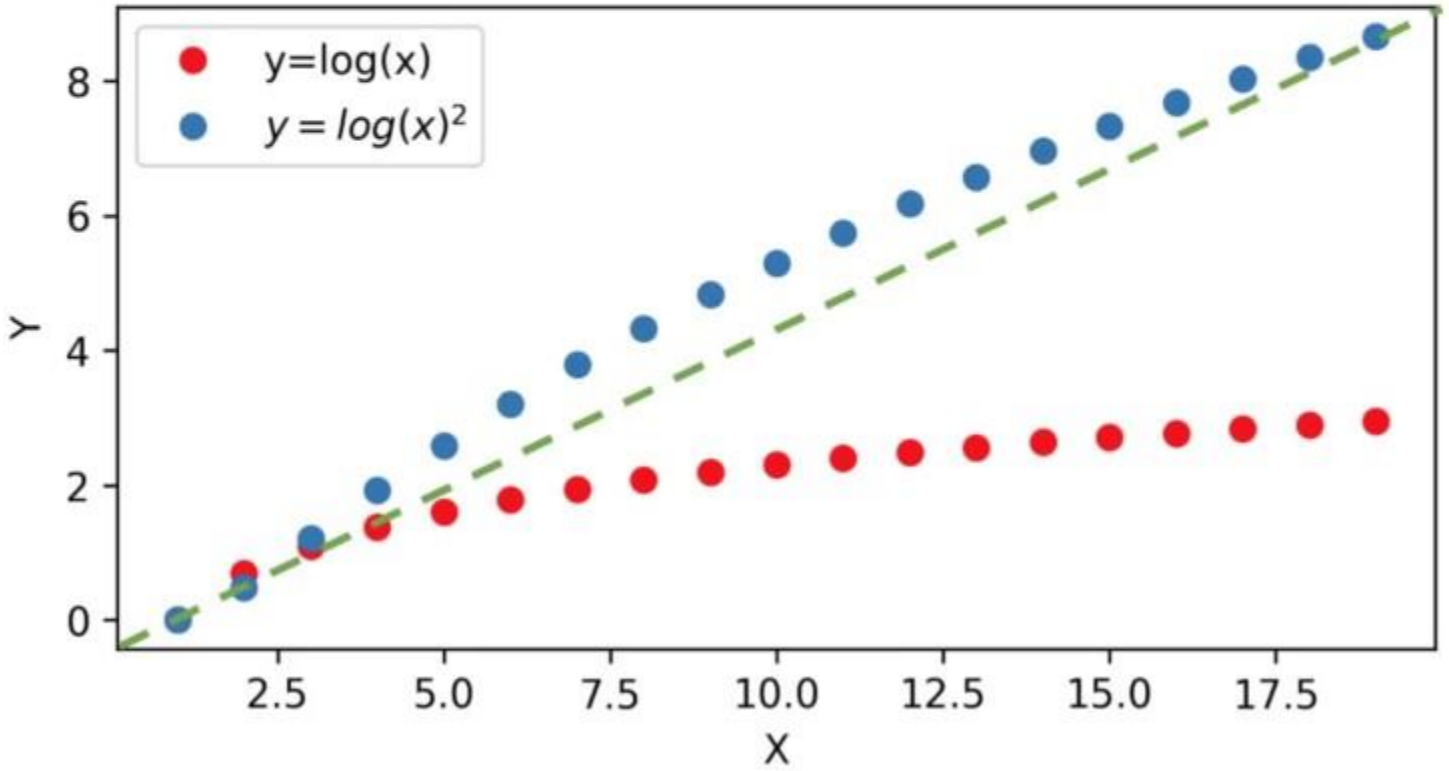
Polynomial Regression

- What about the square of this feature?



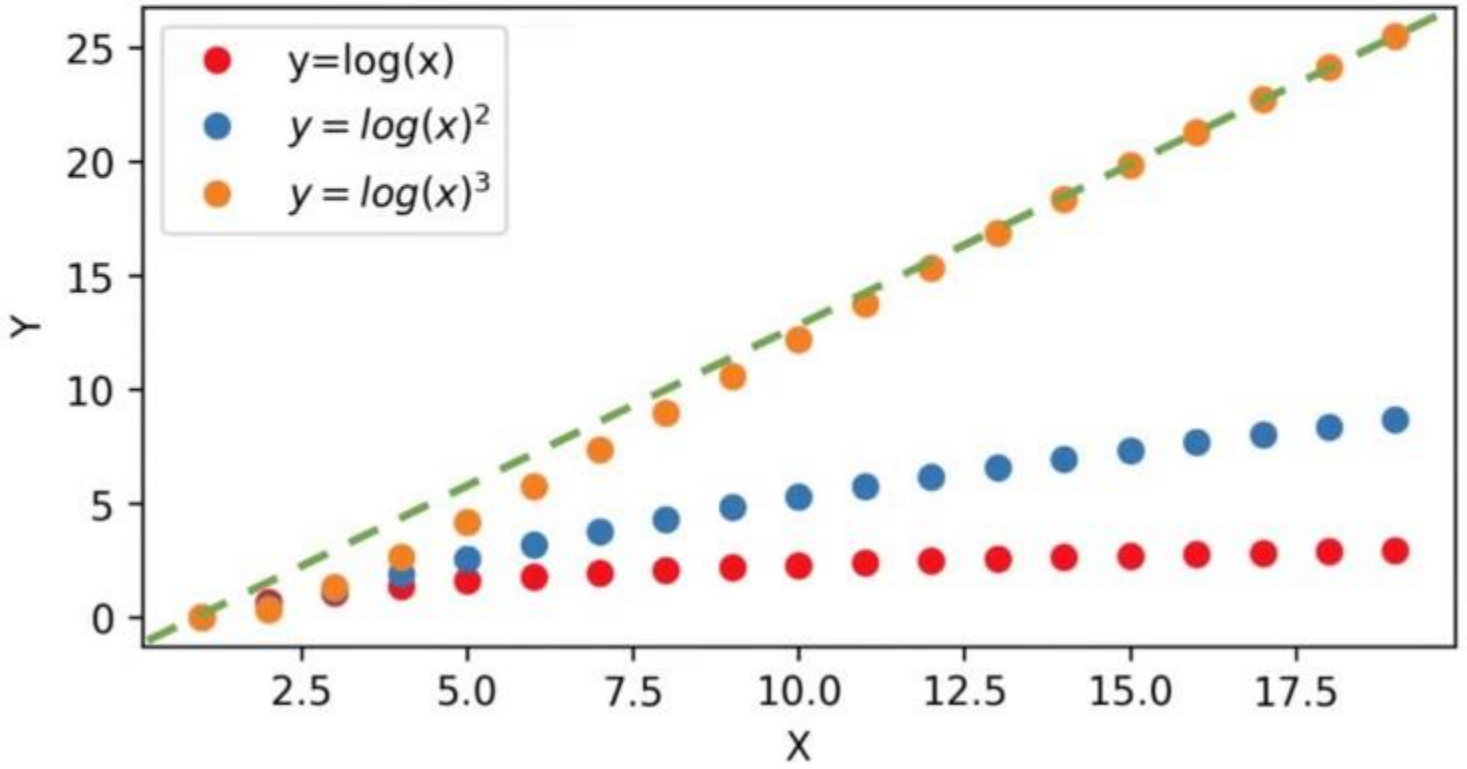
Polynomial Regression

- 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**

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, X_1, X_2, X_1^2, X_1X_2, X_2^2**

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, X_1, X_2, X_1^2, X_1X_2, X_2^2**
- Example if row was X_1=2 and X_2=3
  - **1, 2, 3, 4, 6, 9**

Polynomial Regression

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

Polynomial Regression

Creating Polynomial Features

02-Polynomial-Regression [LEC5].ipynb

Polynomial Regression

Training and Evaluating Model

02-Polynomial-Regression [LEC5].ipynb

Bias-Variance Trade Off

Overfitting versus Underfitting

Overfitting and Underfitting

- We have seen that a higher order polynomial model performs significantly better than a standard linear regression model.
- But how can we choose the optimal degree for the polynomial?
- What trade-offs are we to consider as we increase model complexity?

Overfitting and Underfitting

- In general, increasing model complexity in search for better performance leads to a **Bias-Variance trade-off**.
- We want to have a model that can generalize well to new unseen data, but can also account for variance and patterns in the known data.

Overfitting and Underfitting

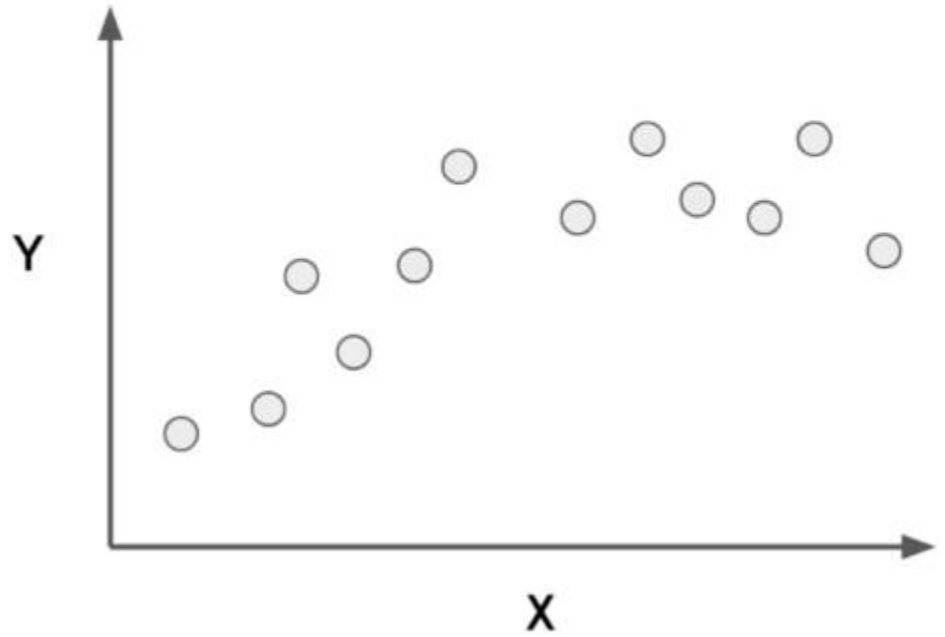
- Extreme bias or extreme variance both lead to bad models.
- We can visualize this effect by considering a model that underfits (high bias) or a model that overfits (high variance).
- Let's start with a model that overfits to a dataset...

Overfitting and Underfitting

- **Overfitting**
  - The model fits too much to the noise from the data.
  - This often results in **low error on training sets but high error on test/validation sets.**

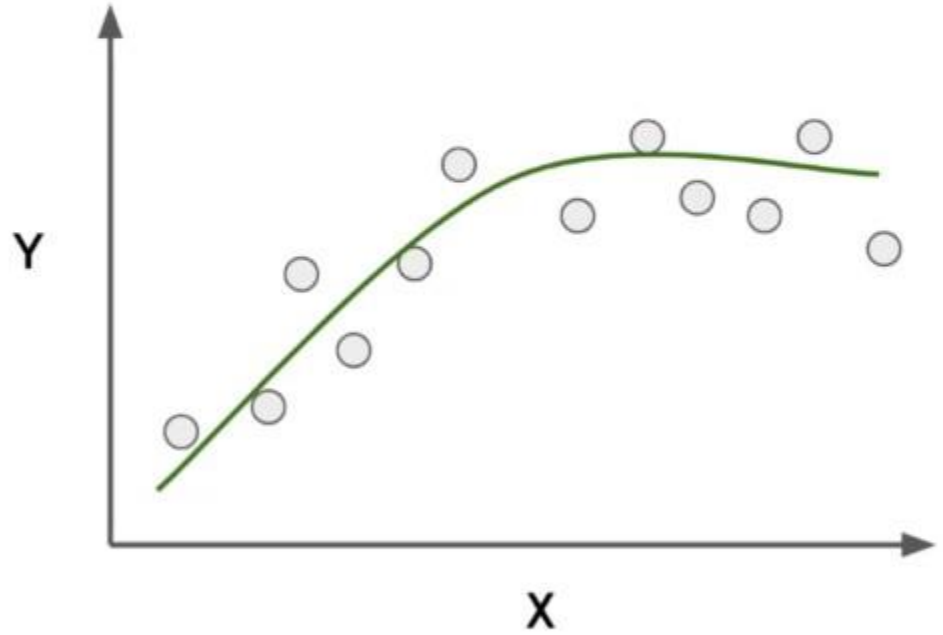
Overfitting and Underfitting

Data



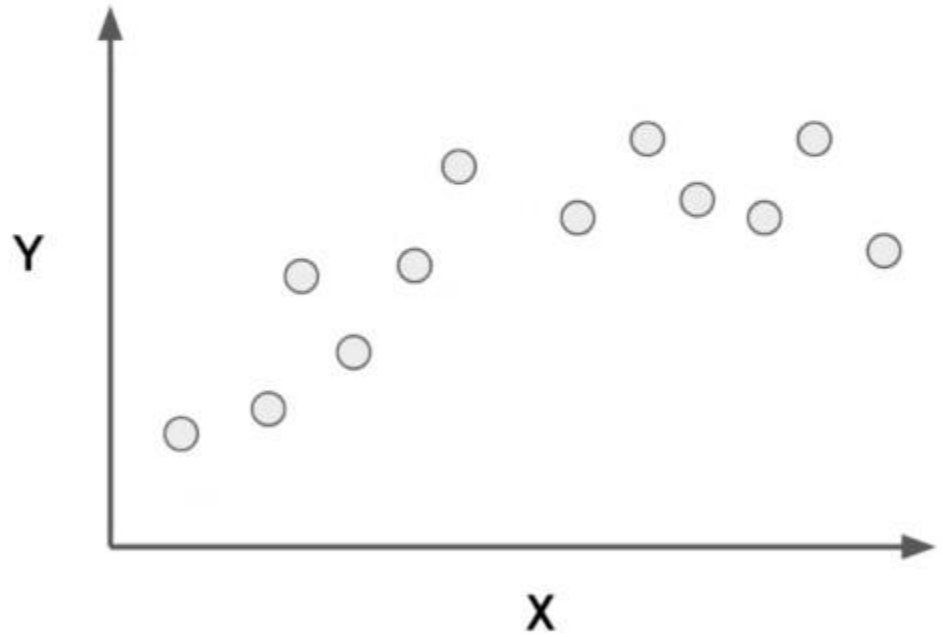
Overfitting and Underfitting

Good Model



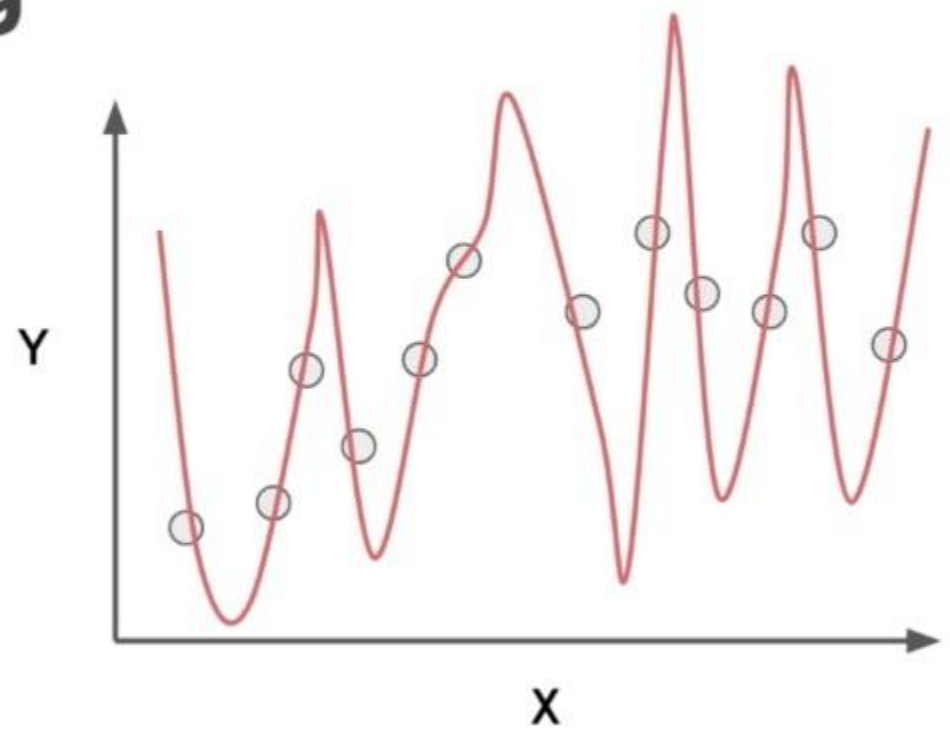
Overfitting and Underfitting

- **Overfitting**



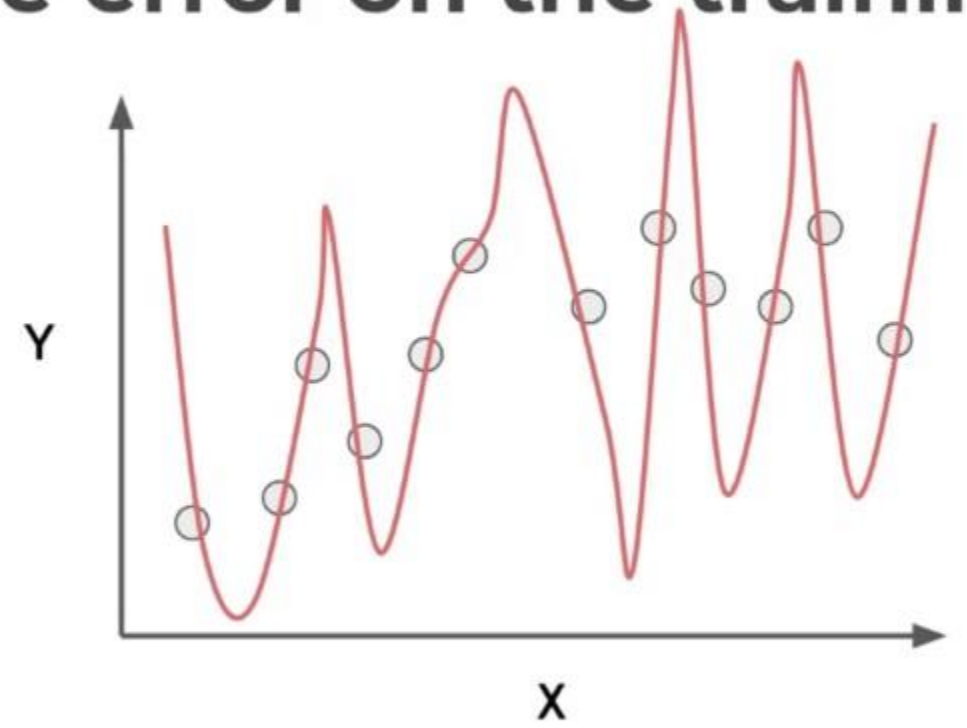
Overfitting and Underfitting

- **Overfitting**



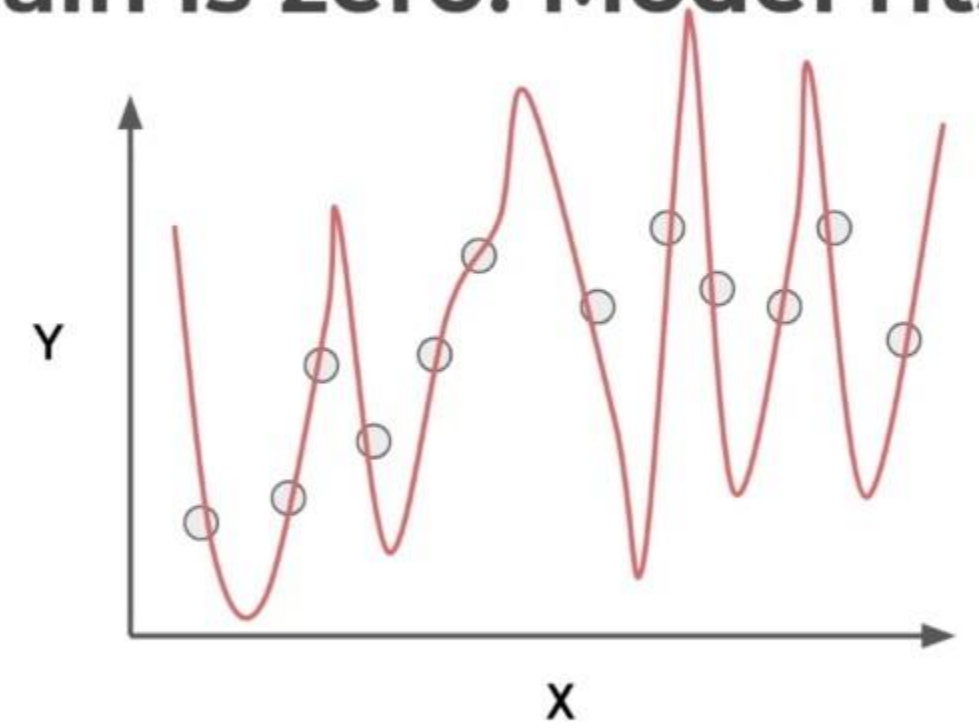
Overfitting and Underfitting

- What is the error on the training data here?



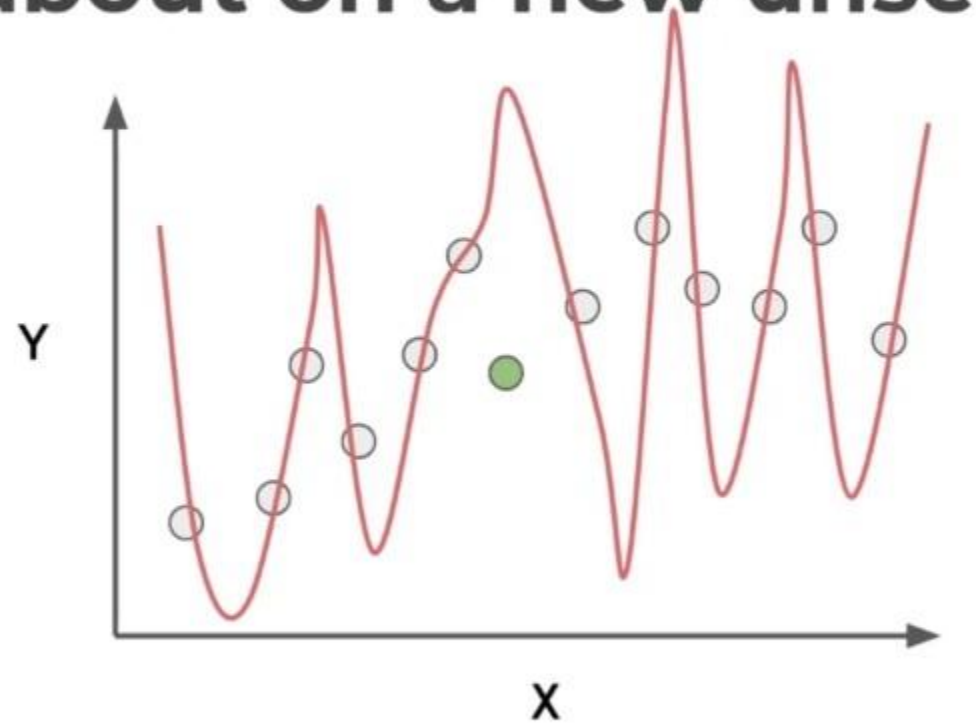
Overfitting and Underfitting

- Error on train is zero! Model fits perfectly!



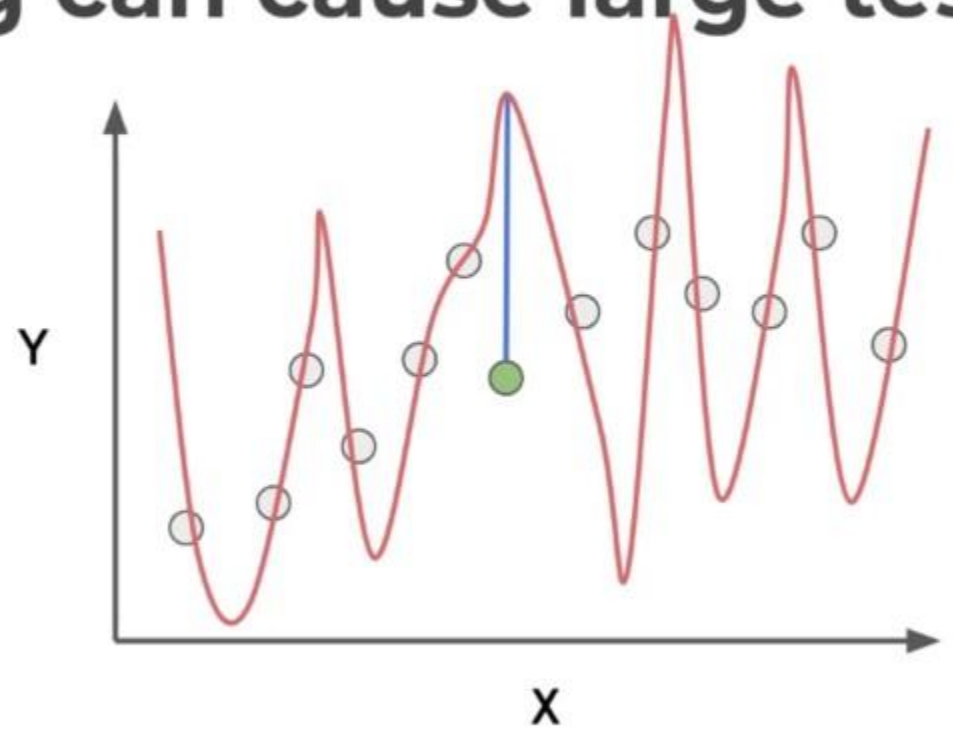
Overfitting and Underfitting

- But what about on a new unseen data point?



Overfitting and Underfitting

- **Overfitting can cause large test errors!**



Overfitting and Underfitting

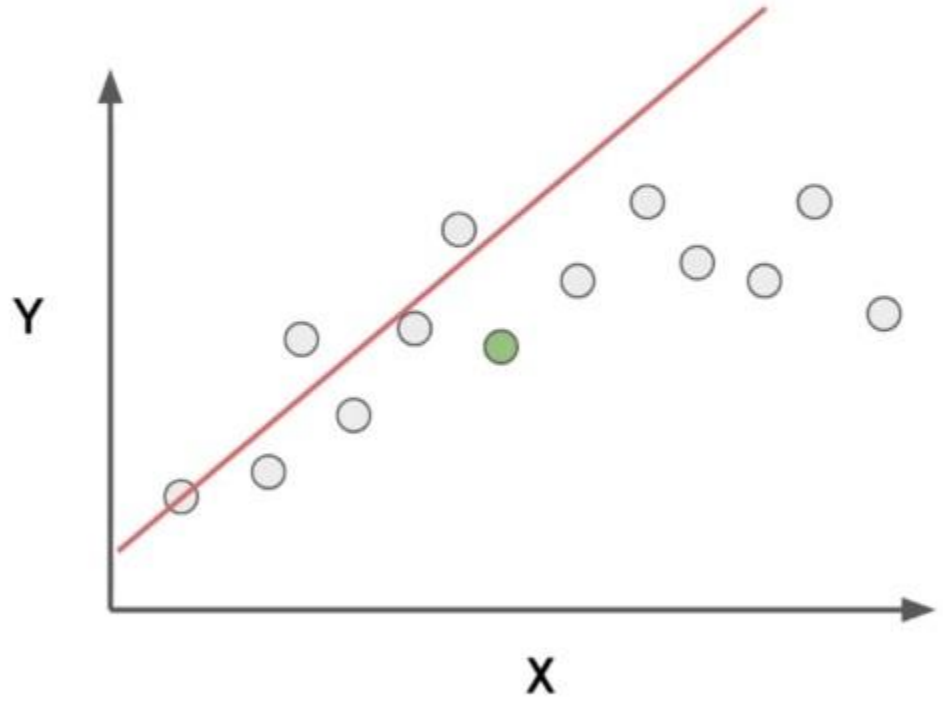
- **Overfitting**
  - Model is fitting too much to noise and variance in the training data.
  - Model will perform very well on training data, but have poor performance on new unseen data.

Overfitting and Underfitting

- **Underfitting**
  - Model does not capture the underlying trend of the data and does not fit the data well enough.
  - Low variance but high bias.
  - Underfitting is often a result of an excessively simple model.

Overfitting and Underfitting

Underfitting



Overfitting and Underfitting

- **Underfitting**
  - Model has high bias and is generalizing too much.
  - Underfitting can lead to poor performance in both training and testing data sets.

Overfitting and Underfitting

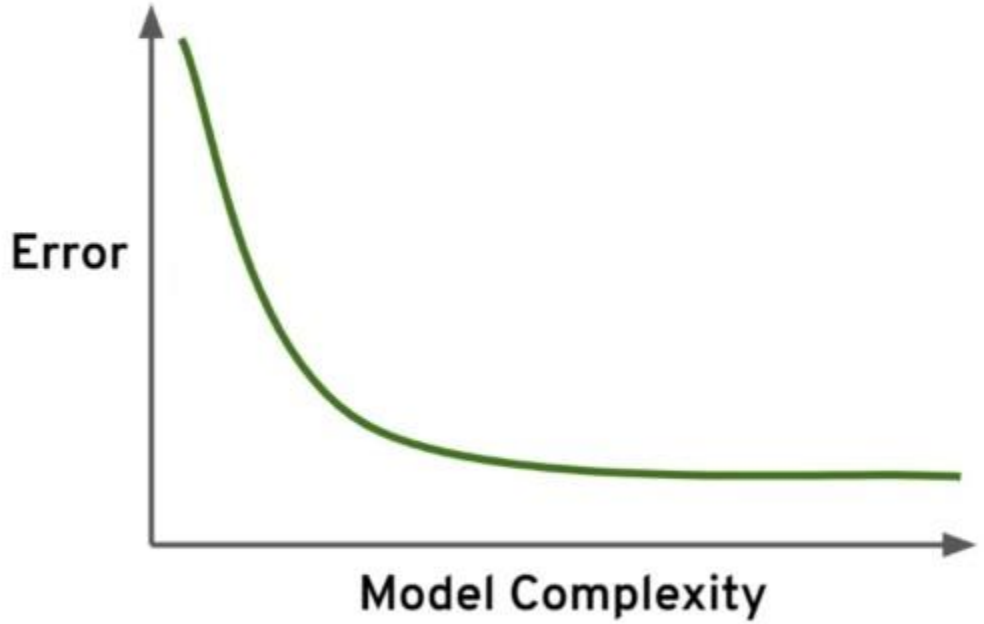
- **Overfitting versus Underfitting**
  - Overfitting can be harder to detect, since good performance on training data could lead to a model that appears to be performing well.

Overfitting and Underfitting

- This data was easy to visualize, but how can we see underfitting and overfitting when dealing with multi dimensional data sets?
- First let's imagine we trained a model and then measured its error versus model complexity (e.g. higher order polynomials).

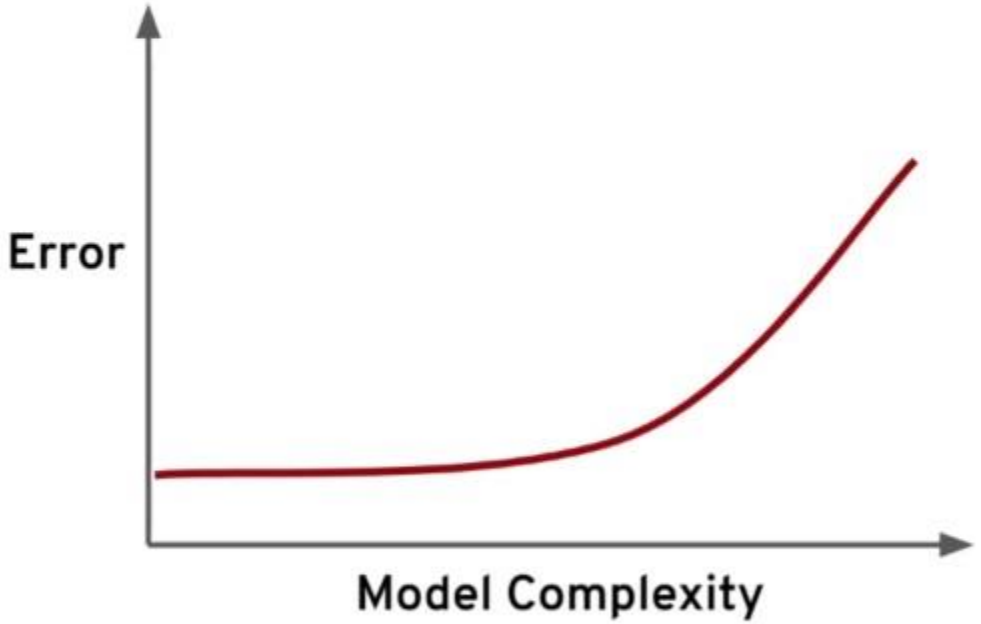
Overfitting and Underfitting

- Good Model



Overfitting and Underfitting

- Bad Model



Overfitting and Underfitting

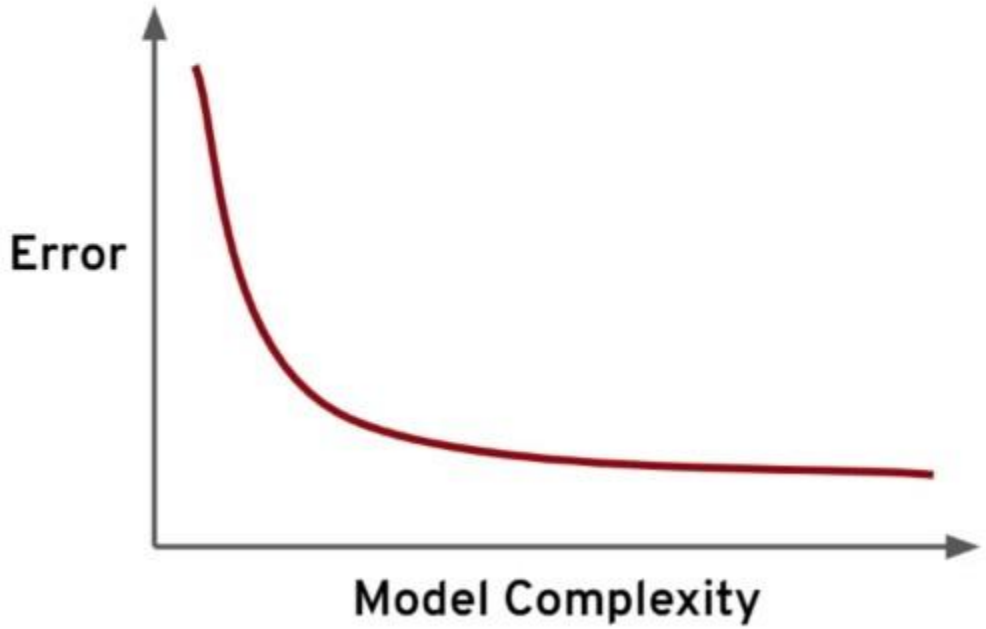
- When thinking about **overfitting** and **underfitting** we want to keep in mind the relationship of model performance on the training set versus the test/validation set.

Overfitting and Underfitting

- Let's imagine we split our data into a **training set** and a **test set**

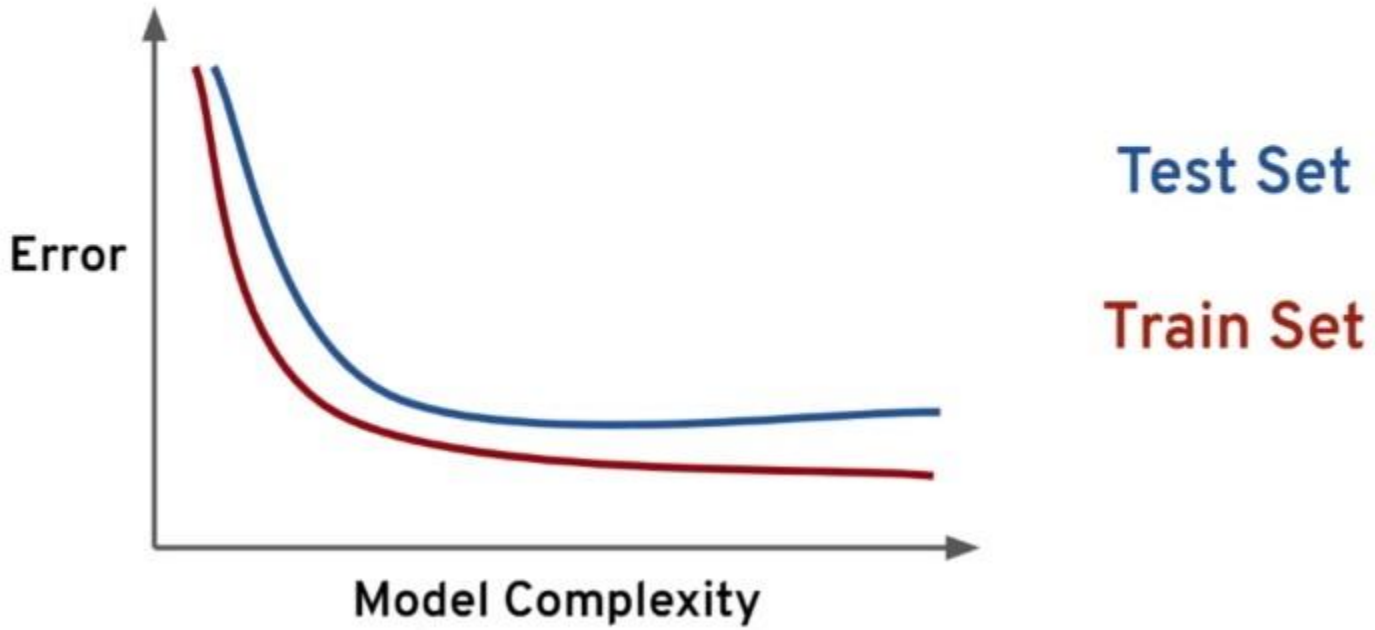
Overfitting and Underfitting

- We first see performance on the **training set**



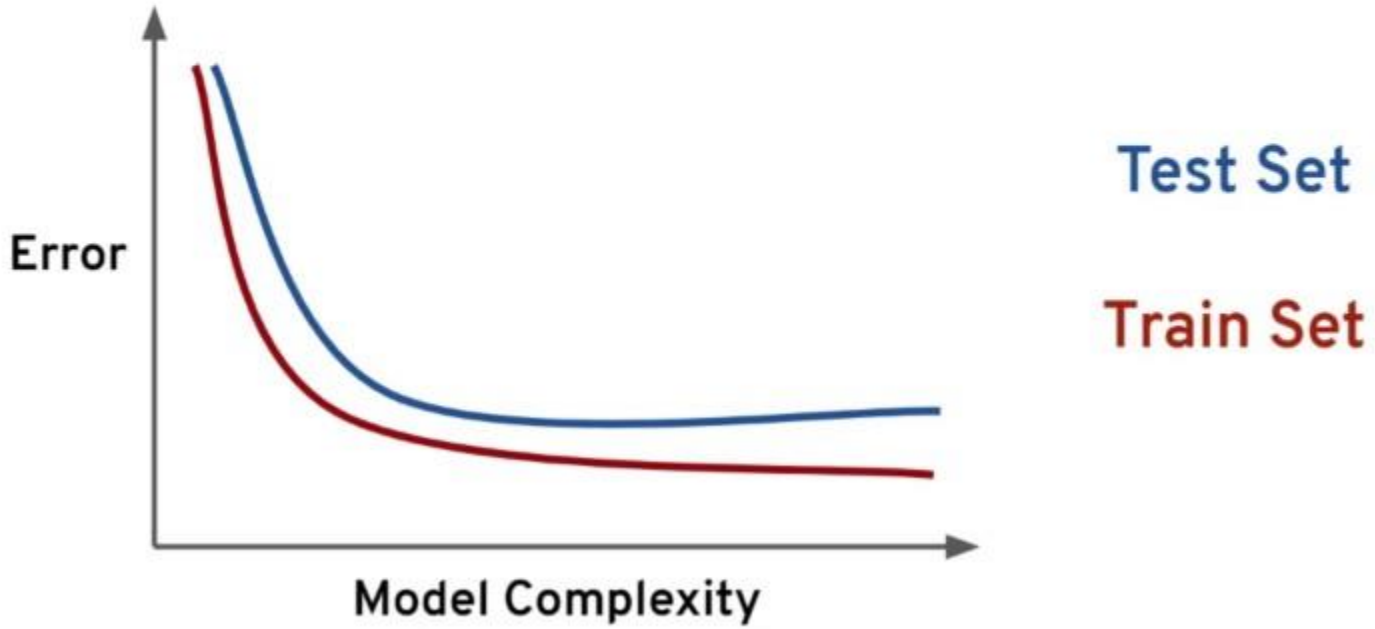
Overfitting and Underfitting

- Next we check performance on the **test set**



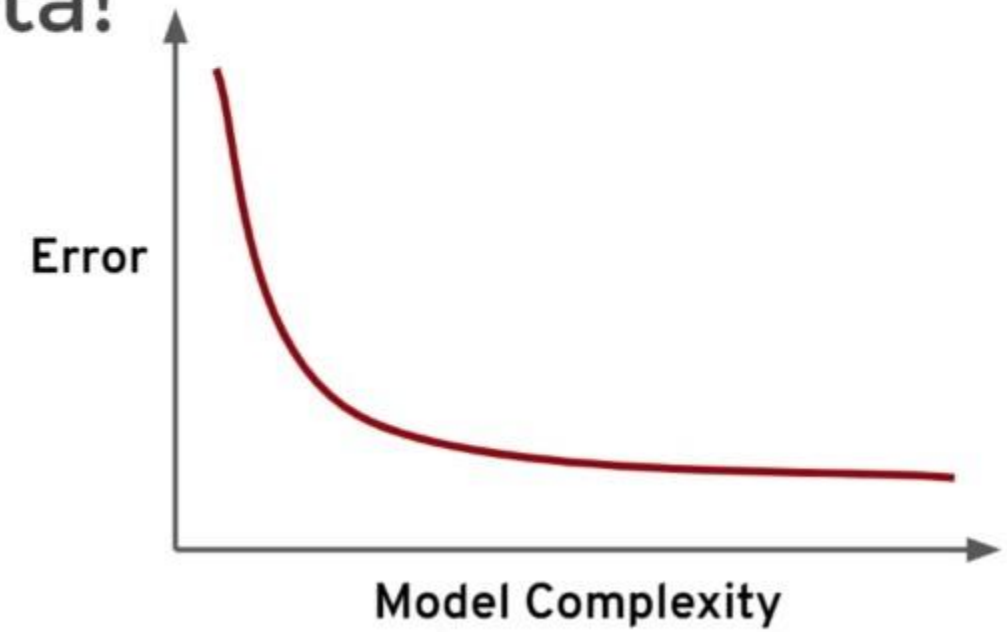
Overfitting and Underfitting

- Ideally the model would perform well on both, with similar behavior.



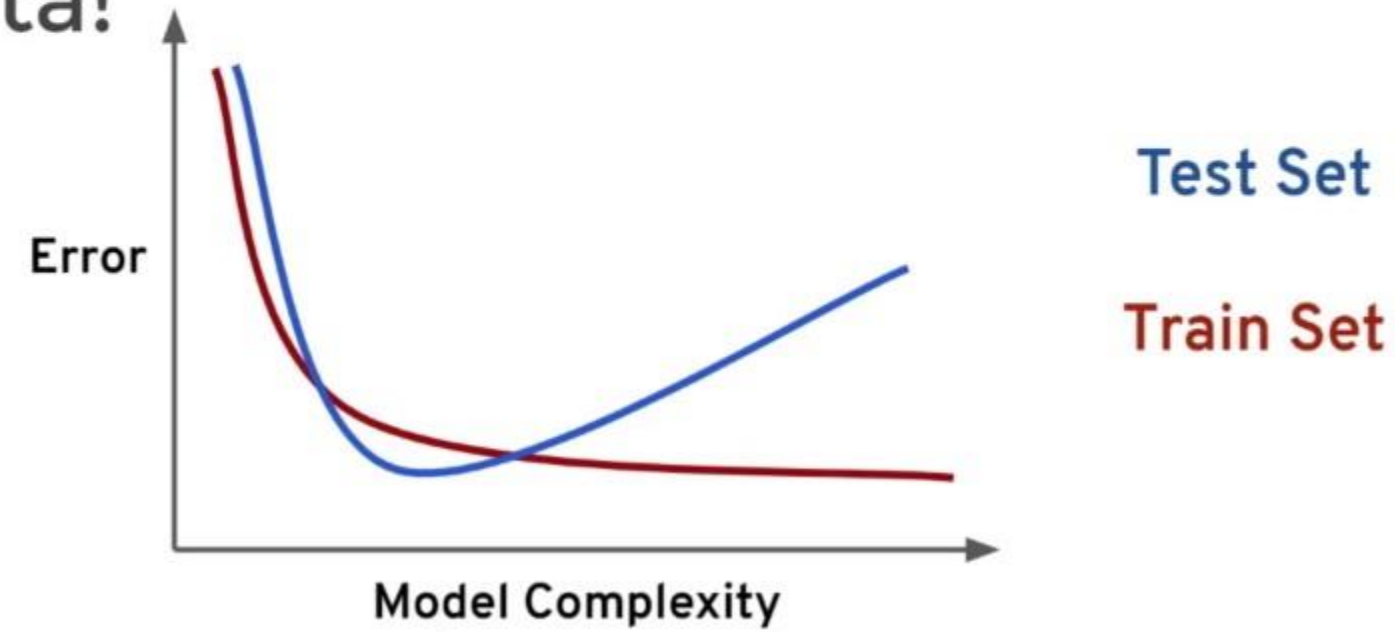
Overfitting and Underfitting

- But what happens if we overfit on the training data? That means we would perform poorly on new test data!



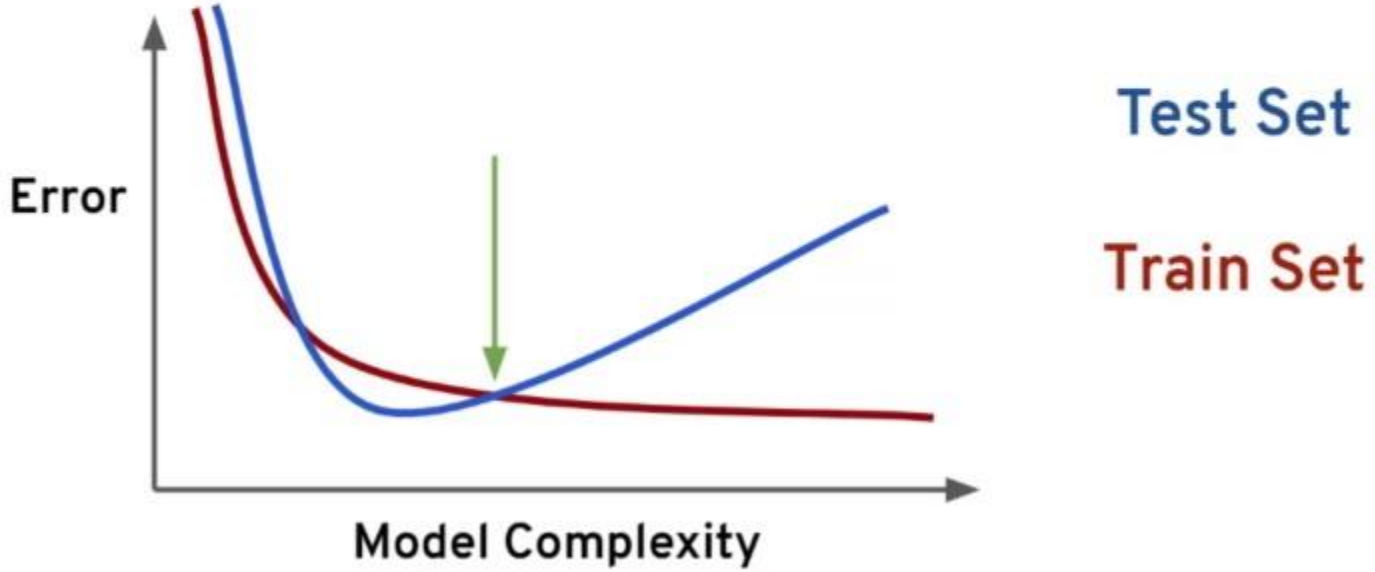
Overfitting and Underfitting

- But what happens if we overfit on the training data? That means we would perform poorly on new test data!



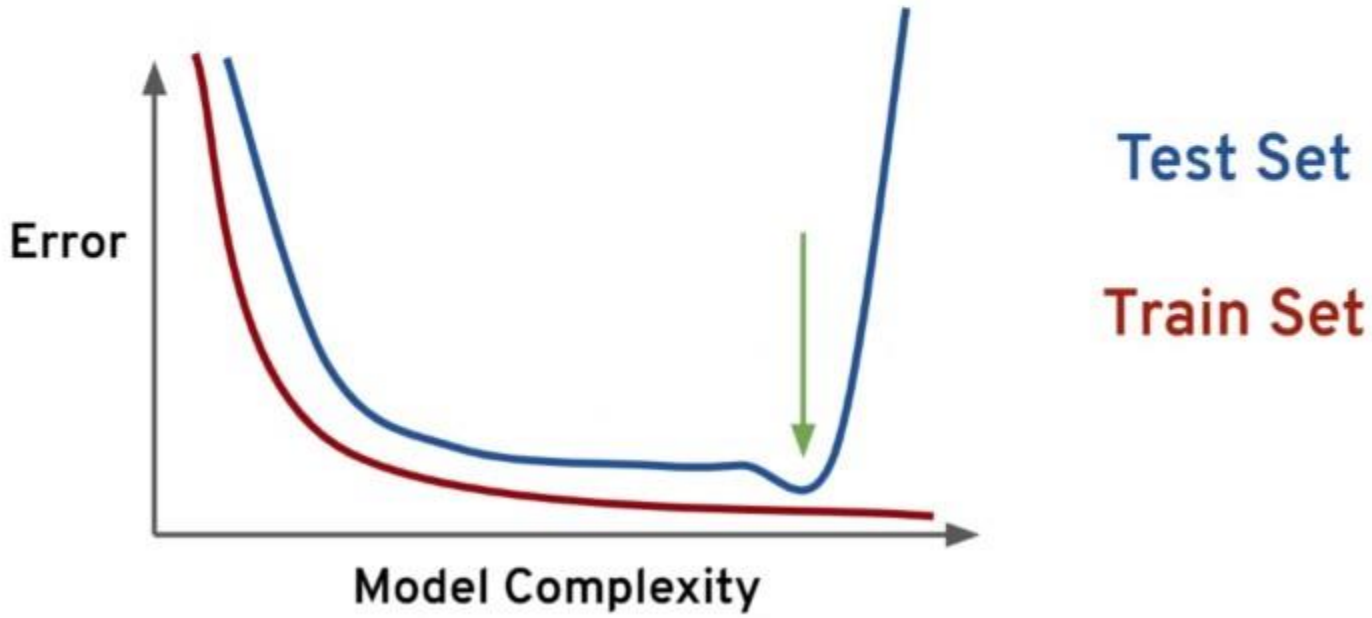
Overfitting and Underfitting

- This is a good indication too much complexity, you should look for the point to determine appropriate values!



Overfitting and Underfitting

- For certain algorithms this test error jump can be sudden instead of gradual.



Overfitting and Underfitting

- This means when deciding optimal model complexity **and** wanting to fairly evaluate our model's performance, we can consider both the train error and test error to select an ideal complexity.

Overfitting and Underfitting

- In the case of Polynomial Regression, complexity directly relates to degree of the polynomial, but many machine learning algorithms have their own hyperparameters that can increase complexity.

Polynomial Regression

Adjusting Model Parameters

Polynomial Regression

- Let's explore choosing the optimal model complexity (order of polynomial).
- As we previously discussed, we will need to understand error for both training and test data to look out for potential overfitting.

02-Polynomial-Regression [LEC5].ipynb

Polynomial Regression

Model Deployment

02-Polynomial-Regression [LEC5].ipynb