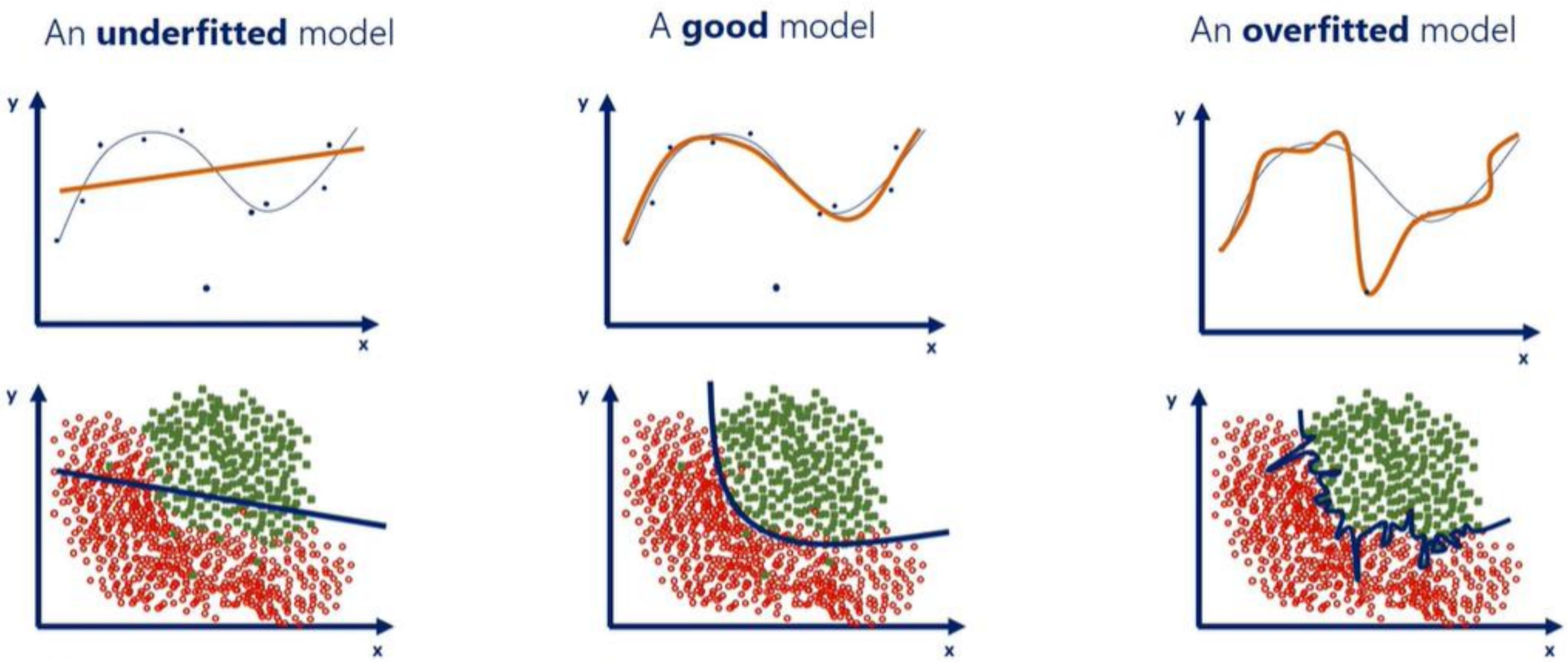
CSC 462: Machine Learning

5.4 Underfitting and Overfitting

5.5 Regularization

Dr. Sultan Alfarhood

Underfitting and Overfitting



Doesn't capture any logic

- High loss
- Low accuracy

Captures the underlying logic of the dataset

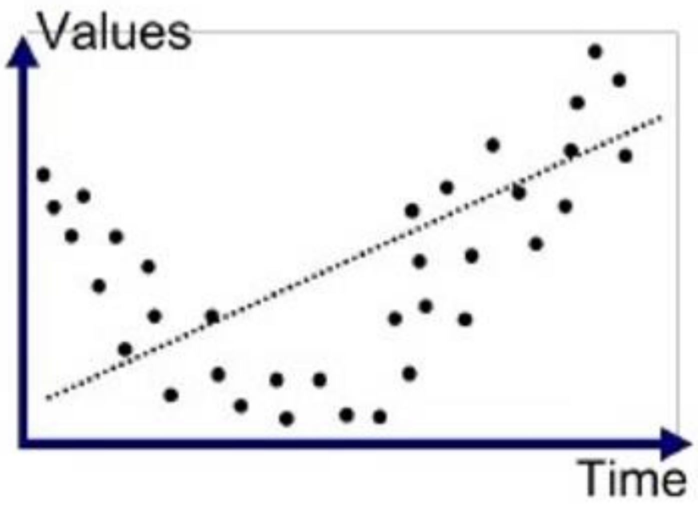
- Low loss
- High accuracy

Captures all the noise, thus "missed the point"

- Low loss
- Low accuracy

Underfitting

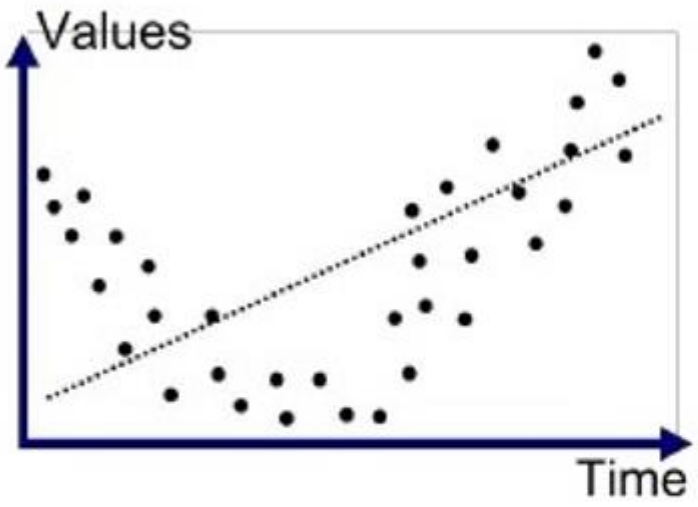
- Underfitting is the inability of the model to predict well the labels of the data it was trained on.



Underfitted

Underfitting

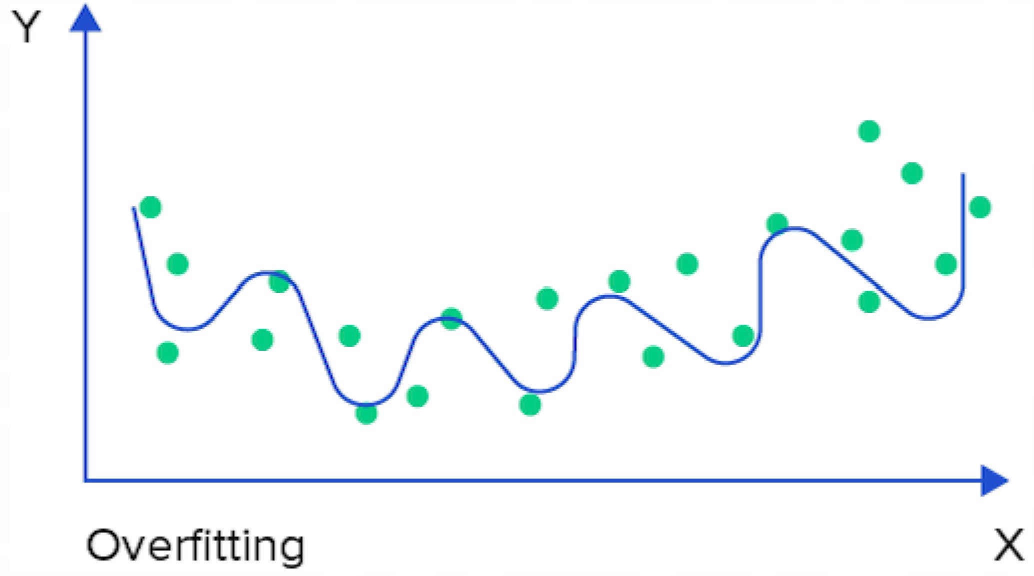
- There could be several reasons for underfitting, the most important of which are:
  - The **model is too simple** for the data
    - For example, a linear model can often underfit
  - The **features** you engineered are **not informative** enough
- The solution of underfitting is to try a more complex model or to engineer features with **higher predictive power.**



Underfitted

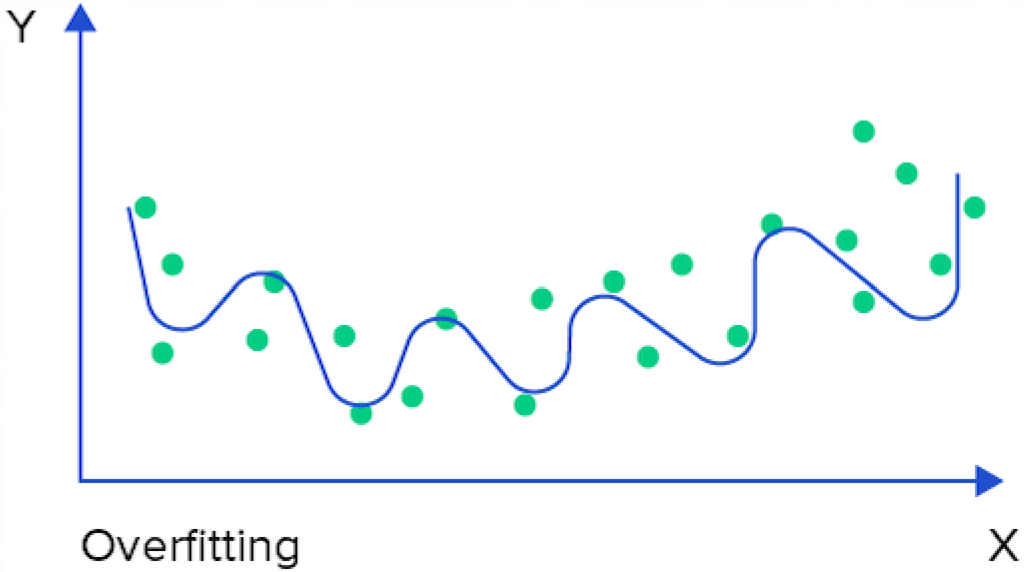
Overfitting

- The model that **overfits** predicts very well the training data but poorly the data from at least one of the two hold-out sets.
- Also referred to as the **problem of high variance.**



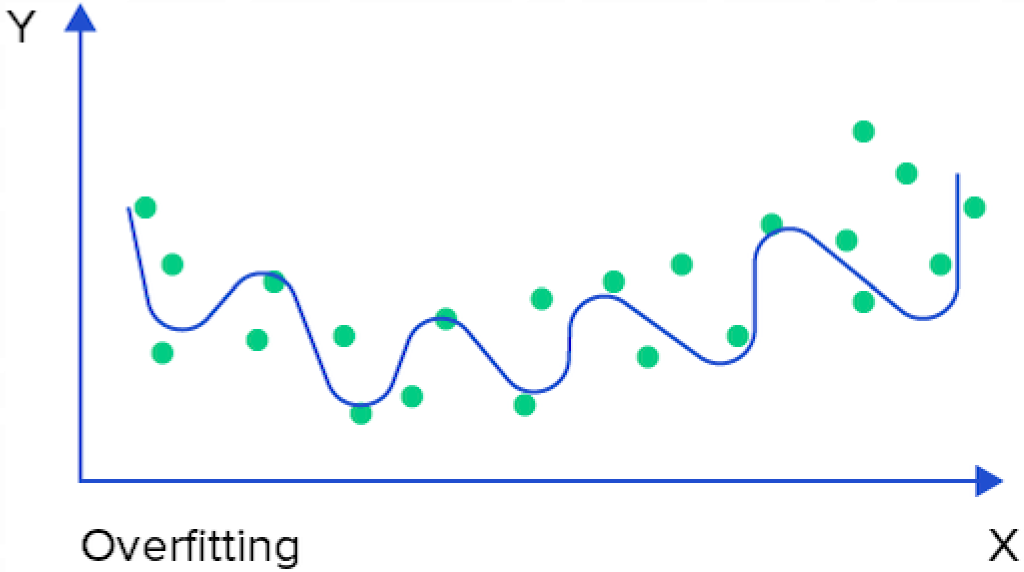
Overfitting

- Several reasons can lead to overfitting, the most important of which are:
  - The model is **too complex** for the data
    - For example, a very tall decision tree or a very deep or wide neural network often overfit
  - Too many features but with a small number of training examples



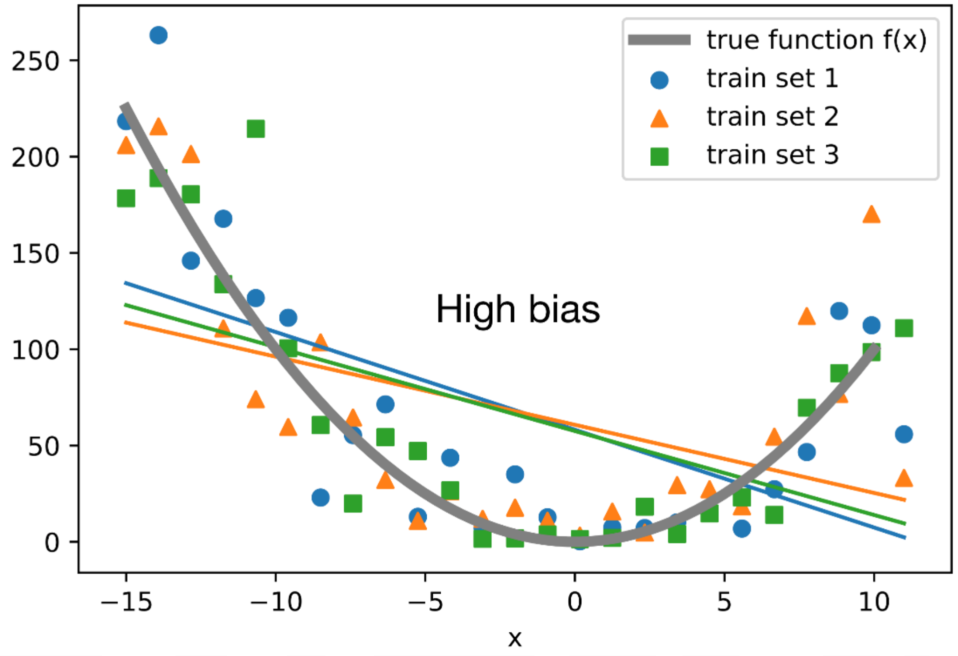
Overfitting

- Several solutions to the problem of overfitting are possible:
  1. Try a **simpler** model
     - Linear instead of polynomial regression
     - SVM with a linear kernel instead of RBF
     - A neural network with fewer layers/units
  2. **Reduce** the **dimensionality** of examples in the dataset
     - For example, by using one of the dimensionality reduction techniques discussed in Chapter 9
  3. Add **more** training **data**, if possible
  4. **Regularize** the model



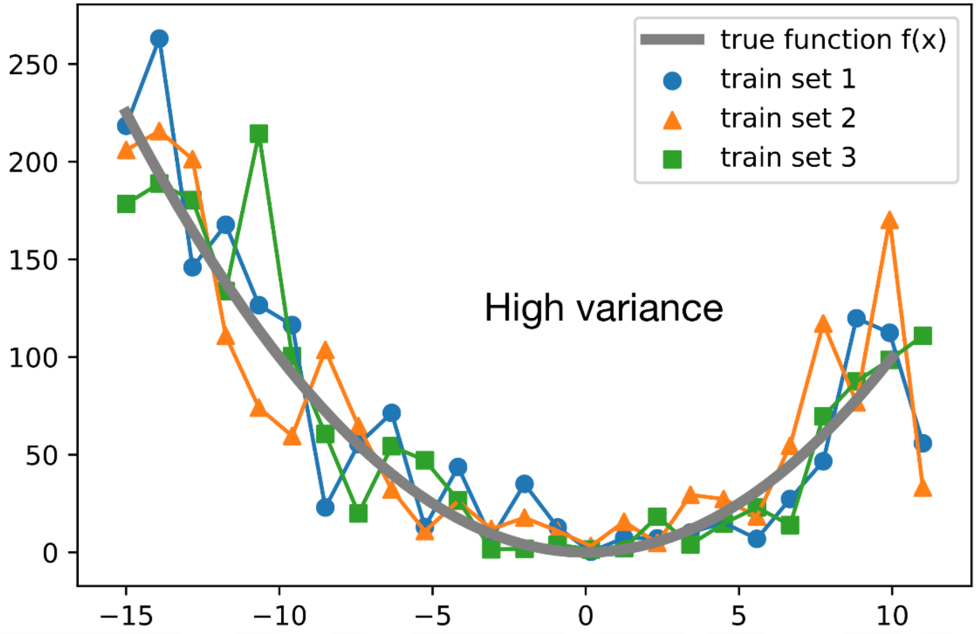
Bias

- **Bias** is the difference between the average prediction of our model and the correct value which we are trying to predict
  - If the model makes many mistakes on the training data, we say that the model has a **high bias** or that the model **underfits**
  - A model has a **low bias** if it predicts well the labels of the training data.

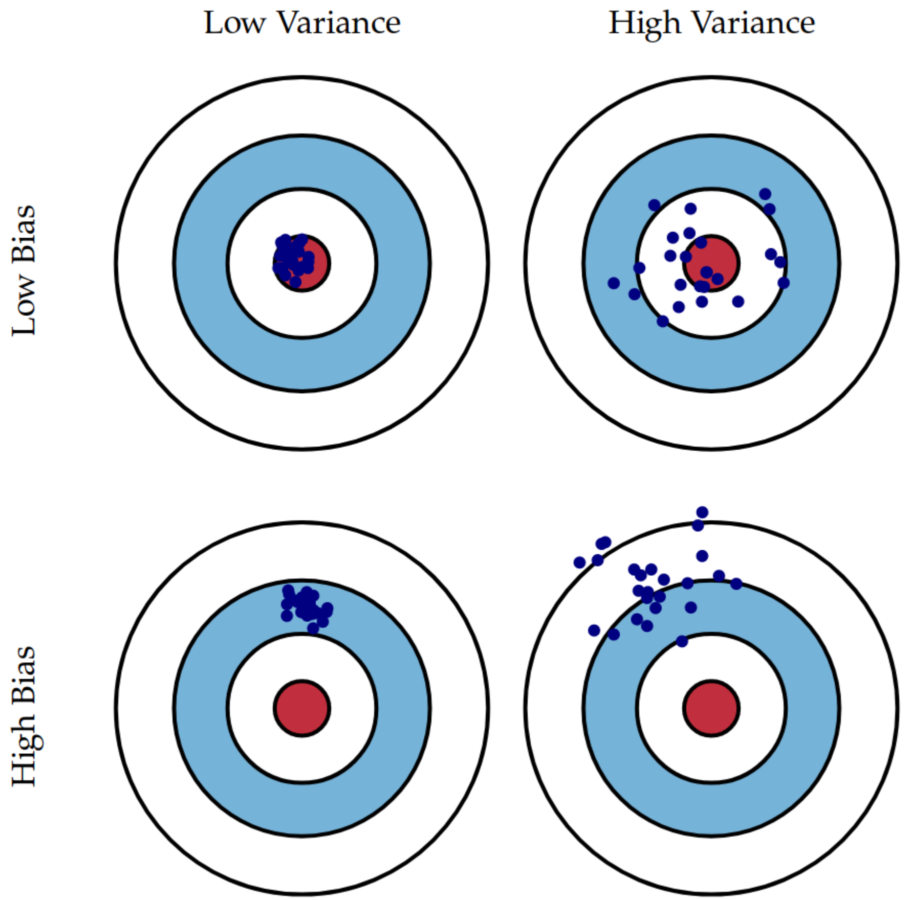


Variance

- The **variance** is an error of the model due to its sensitivity to small fluctuations in the training set
  - Variance describes how much a model changes when trained using different portions of a data set.



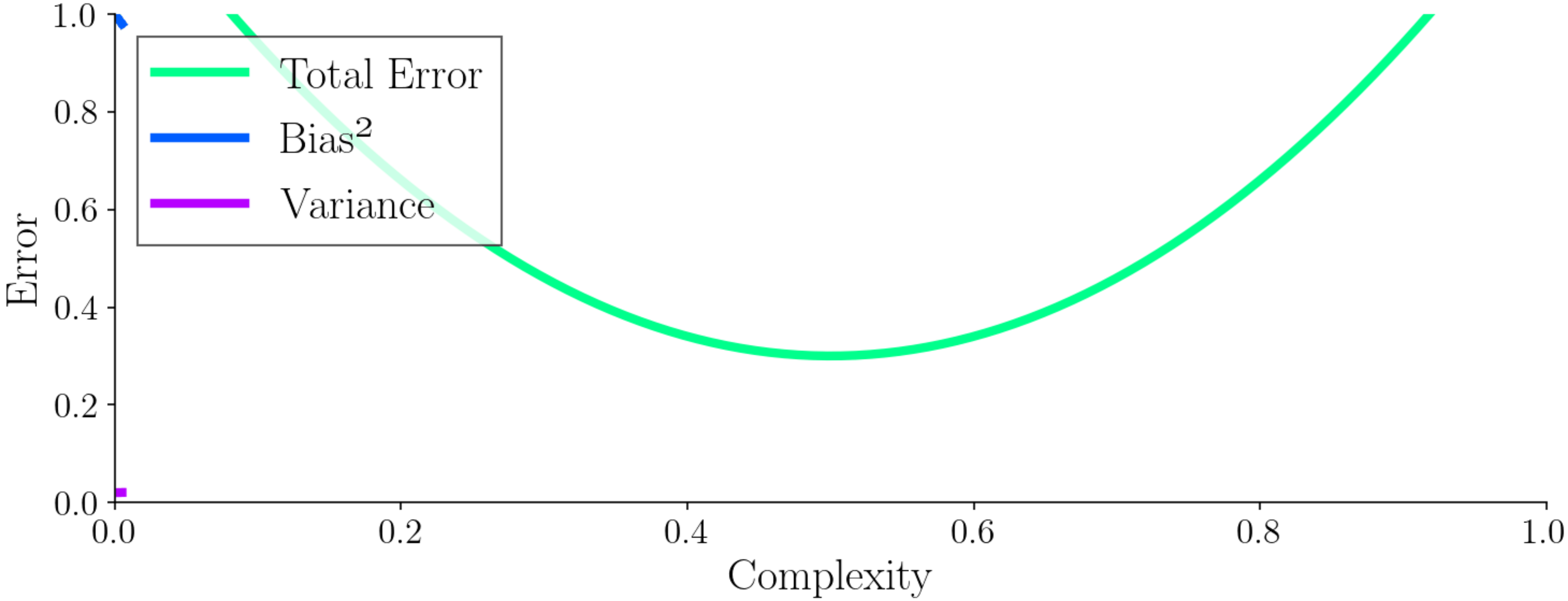
Bias & Variance



Regularization

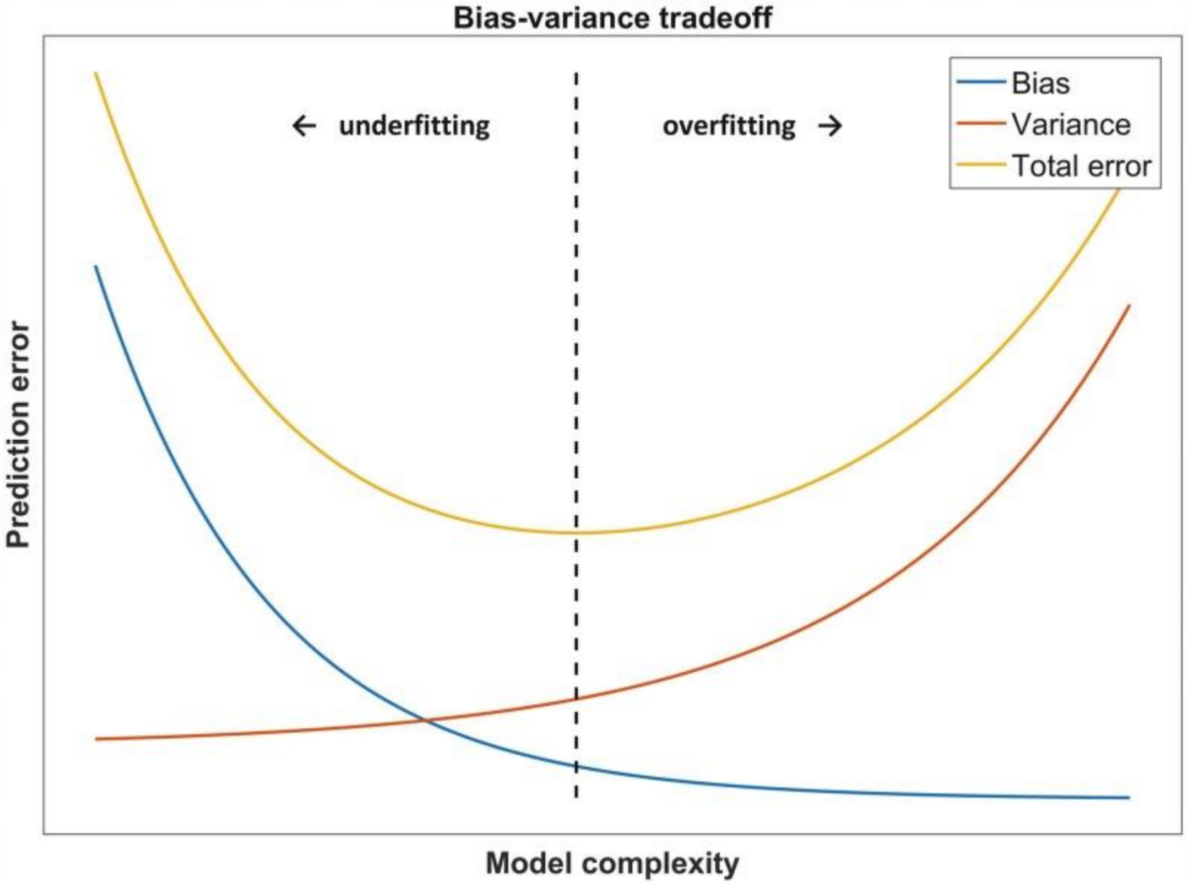
- **Regularization** is a general term that includes methods that **force the learning algorithm to build a less complex model**
  - A penalty is imposed on models, which are very complex
- In practice, that often leads to slightly higher bias but significantly reduces the variance.
  - This problem is known in the literature as the **bias-variance tradeoff**

Bias-variance Tradeoff (Animated)



- **Bias** is the difference between the average prediction of our model and the correct value which we are trying to predict
- **Variance** is an error of the model due to its sensitivity to small fluctuations in the training set

Bias-variance Tradeoff



- **Bias** is the difference between the average prediction of our model and the correct value which we are trying to predict
- **Variance** is an error of the model due to its sensitivity to small fluctuations in the training set

Regularization

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - f_{w,b}(x_i)\right)^2$$

- Regularization is the most widely used approach to prevent overfitting
- The two most widely used types of regularization are
  - L1 regularization (Lasso Regression)
  - L2 regularization (Ridge Regression)

$$\text{L1 objective} = \left(\frac{1}{N}\sum_{i=1}^{N}\left(y_i - f_{w,b}(x_i)\right)^2\right) + \left(\lambda \sum_{j=1}^{D}\left|w^j\right|\right)$$

$$\text{L2 objective} = \left(\frac{1}{N}\sum_{i=1}^{N}\left(y_i - f_{w,b}(x_i)\right)^2\right) + \left(\lambda \sum_{j=1}^{D}\left(w^j\right)^2\right)$$

λ is the regularization coefficient which determines how much regularization we want.

L1 & L2 Regularization

- To create a regularized model, we modify the objective function by adding a penalizing term whose value is higher when the model is more complex.
- The **key difference** between these techniques is that L1 shrinks the less important feature's coefficient to zero thus, removing some feature altogether
  - So, this works well for **feature selection** in case we have a huge number of features

L1 regularization (Lasso Regression):

$$\text{L1 objective} = \left(\frac{1}{N}\sum_{i=1}^{N}\left(y_i - f_{w,b}(x_i)\right)^2\right) + \left(\lambda \sum_{j=1}^{D}|w^j|\right)$$

L2 regularization (Ridge Regression):

$$\text{L2 objective} = \left(\frac{1}{N}\sum_{i=1}^{N}\left(y_i - f_{w,b}(x_i)\right)^2\right) + \left(\lambda \sum_{j=1}^{D}\left(w^j\right)^2\right)$$

λ is the regularization coefficient which determines how much regularization we want.

L1 & L2 Regularization

- In practice, L1 regularization produces a sparse model
  - A model that has most of its parameters equal to zero (provided the hyperparameter $\boldsymbol{\lambda}$ is large enough).
- L1 makes feature selection by deciding which features are essential for prediction and which are not.
  - That can be useful in case you want to increase model explainability
- However, if your only goal is to maximize the performance of the model on the hold-out data, then L2 usually gives better results.

L1 & L2 Regularization

Comparison of L1 and L2 regularization	
L1 regularization	***L2 regularization***
Sum of absolute value of weights	Sum of square of weights
Sparse solution	Non-sparse solution
Multiple solutions	One solution
Built-in feature selection	No feature selection
Robust to outliers	Not robust to outliers (due to the square term)

Regularization