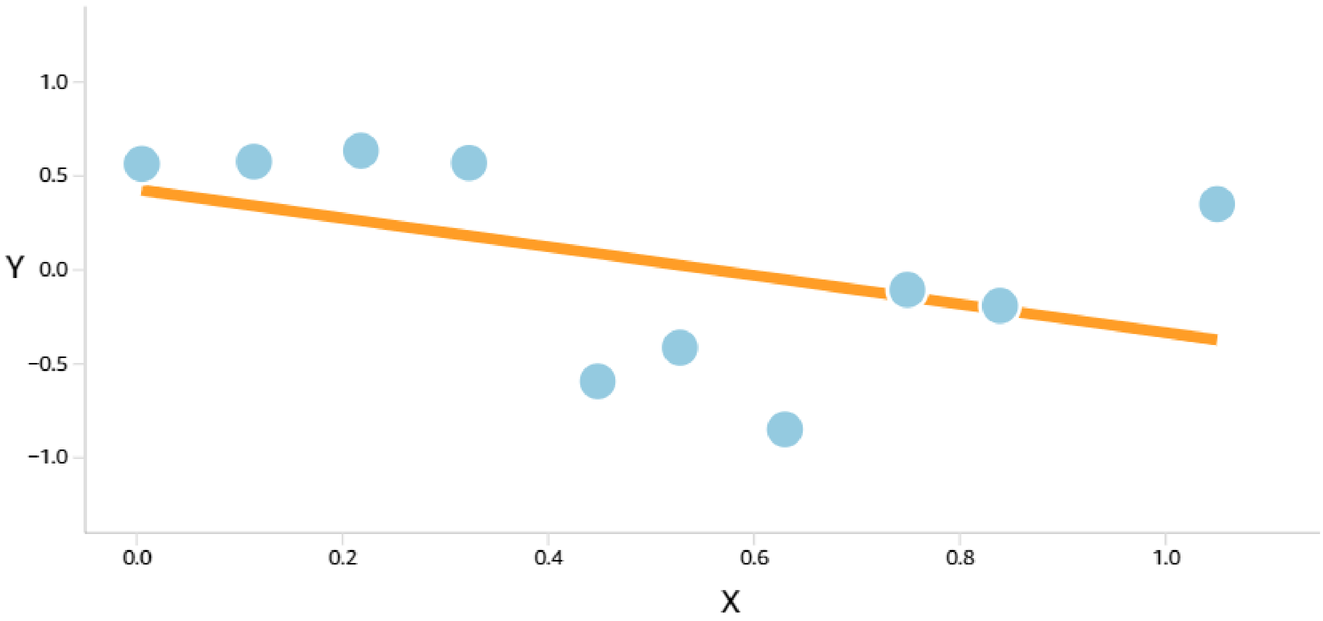
The Bias-Variance Trade-off

- Prediction errors can be decomposed into two main subcomponents of interest: **error from bias, and error from variance.**

- The tradeoff between a model's ability to minimize bias and variance is foundational to training machine learning models.



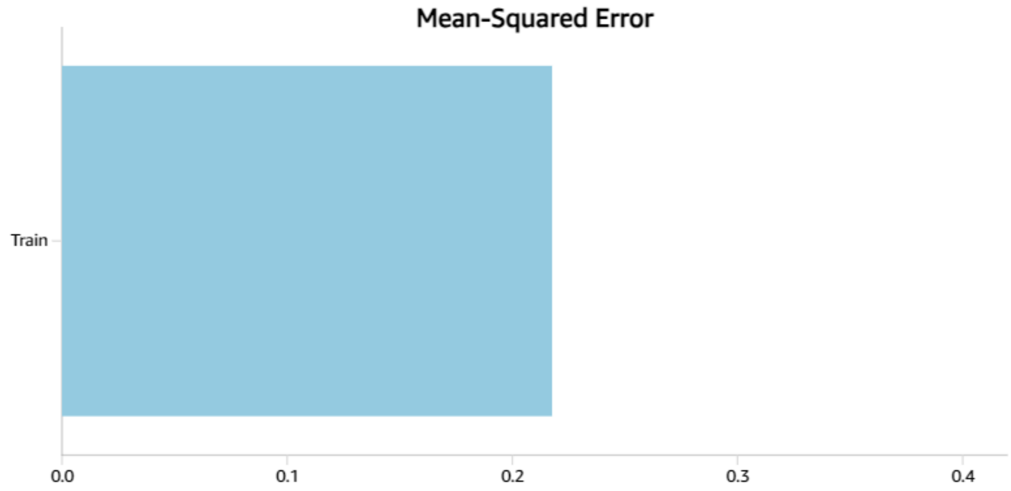
A dataset consisting of features X and labels Y. To generalize this relationship to additional values of X - that will be used to predict future values based on what we’ve already seen before.

Take a very simple approach to modeling the relationship between X and Y by just drawing a line to the general trend of the data.

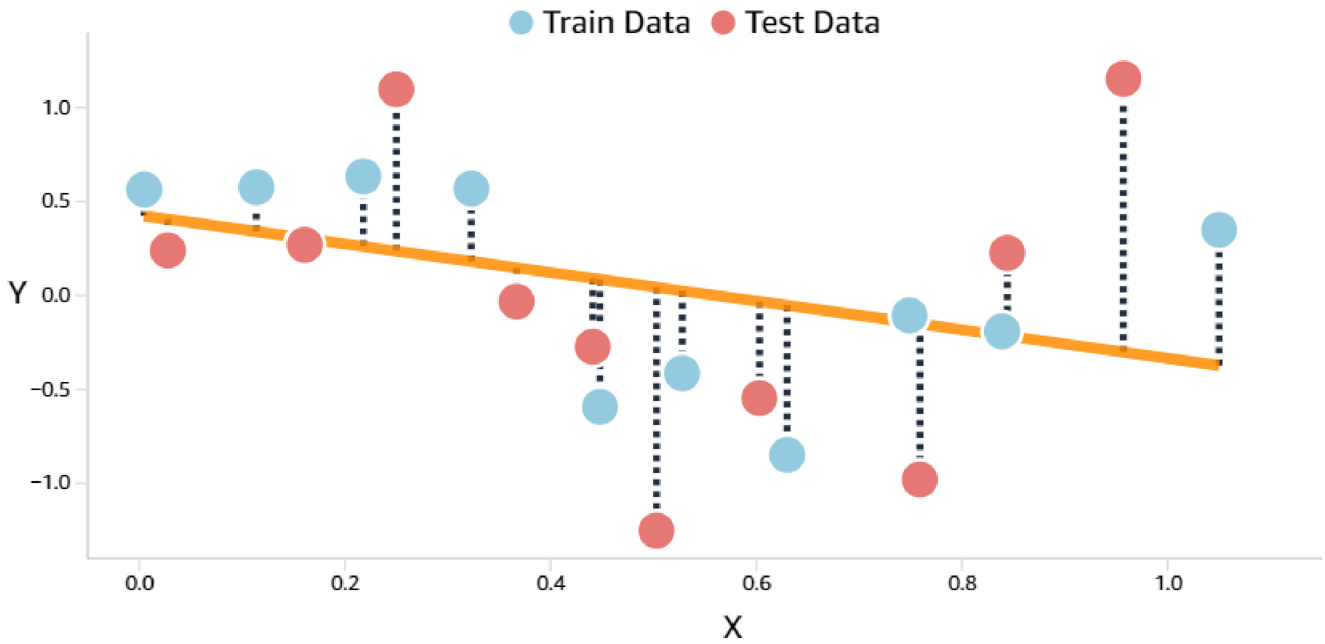
A Simple Model

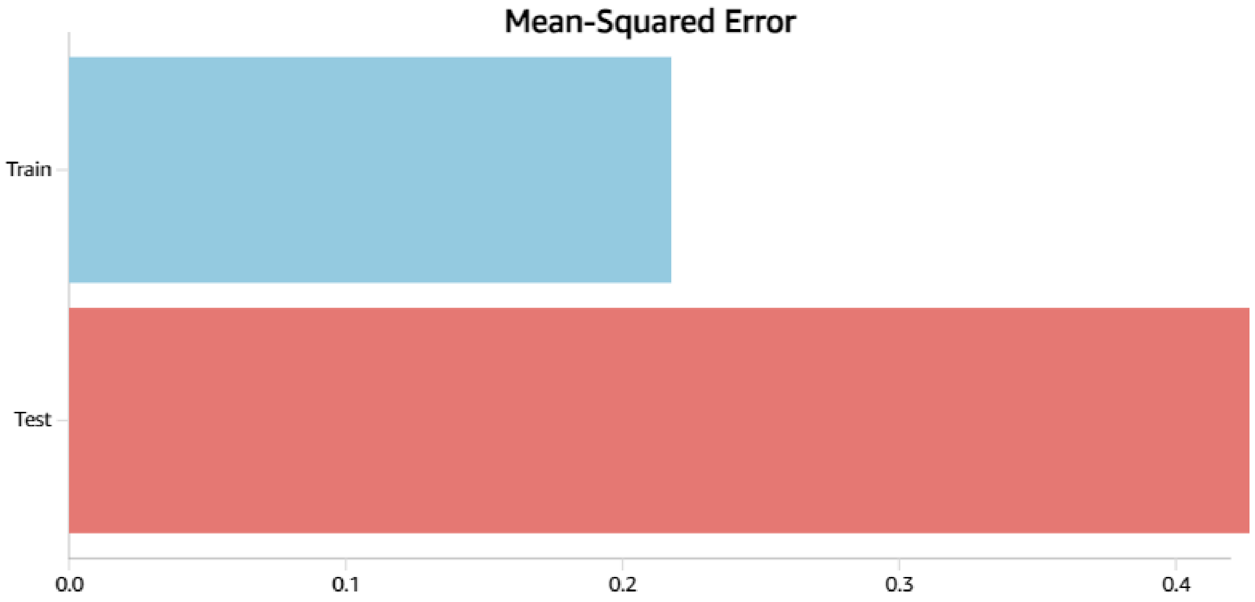
The above simple model isn’t the best at modeling the relationship - clearly there's information in the data that it's failing to capture.

Measure the performance of the model by looking at the **mean-squared error** of its output and the true values (displayed in the bottom barchart). Our model is close to some of the training points, but overall there's definitely room for improvement.



The error on the training data is important for model tuning, but what we really care about is how it performs on data we haven't seen before, called test data.


● Train Data ● Test Data
Y
1.0
0.5
0.0
−0.5
−1.0
0.0
0.2
0.4
0.6
0.8
1.0
X


Mean-Squared Error
Train
Test
0.0
0.1
0.2
0.3
0.4

Low Complexity & Underfitting

The test error is even higher than the train error.

In this case, we say that our model is underfitting the data: our model is so simple that it fails to adequately capture the relationships in the data. The high test error is a direct result of the lack of complexity of our model.

An underfit model is one that is too simple to accurately capture the relationships between its features X and label Y.

Bias (Underfitting)

- Bias refers to the error due to overly simplistic assumptions in the learning algorithm.
- A model with high bias tends to underfit the data, meaning it oversimplifies the underlying patterns.
- This leads to poor predictive performance as the model cannot capture the complexity of the real-world problem.

Example of High Bias:

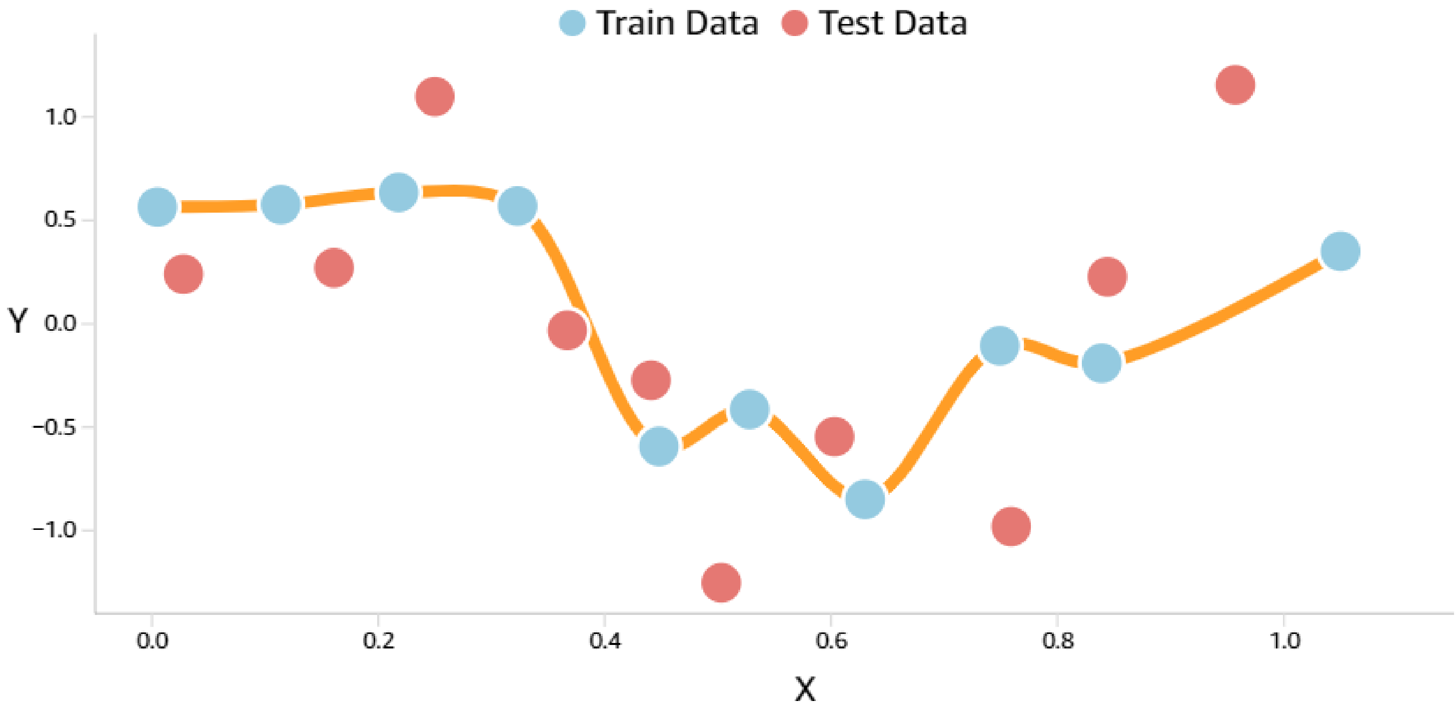
Suppose we are training a linear regression model to predict housing prices. If the model assumes that the relationship between the features (e.g., square footage, number of bedrooms) and the price is strictly linear, it may perform poorly when the true relationship is more complex.

A Complex Model

Previous model performed poorly because it was too simple.

Train a model that predicts every point in our training data perfectly.

Now training error is zero.



High Complexity & Overfitting

Training error from the model is effectively zero, but the error on test data is high.

What gives?

Unsurprisingly, model is too complicated. It overfits the data. Instead of learning the true trends underlying our dataset, it memorized noise and, as a result, the model is not generalizable to datasets beyond its training data.

Overfitting refers to the case when a model is so specific to the data on which it was trained that it is no longer applicable to different datasets.

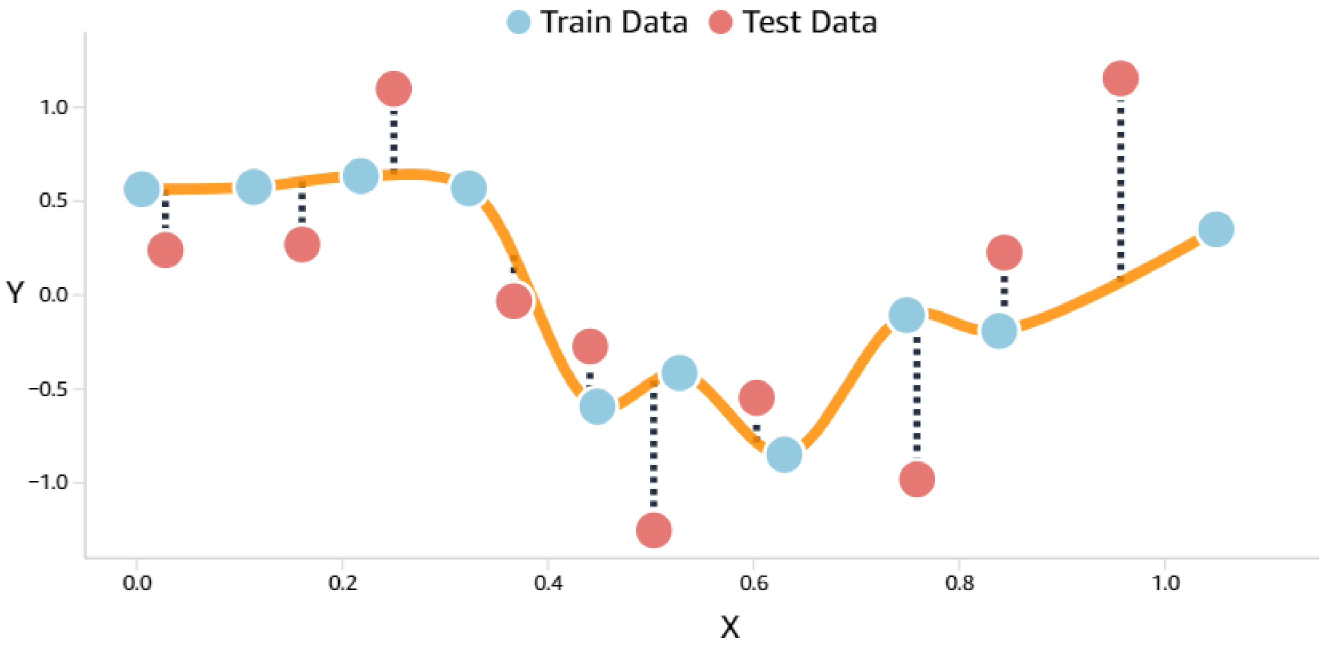
In situations where training error is low but test error is high, the model is overfitted.

Variance (Overfitting)

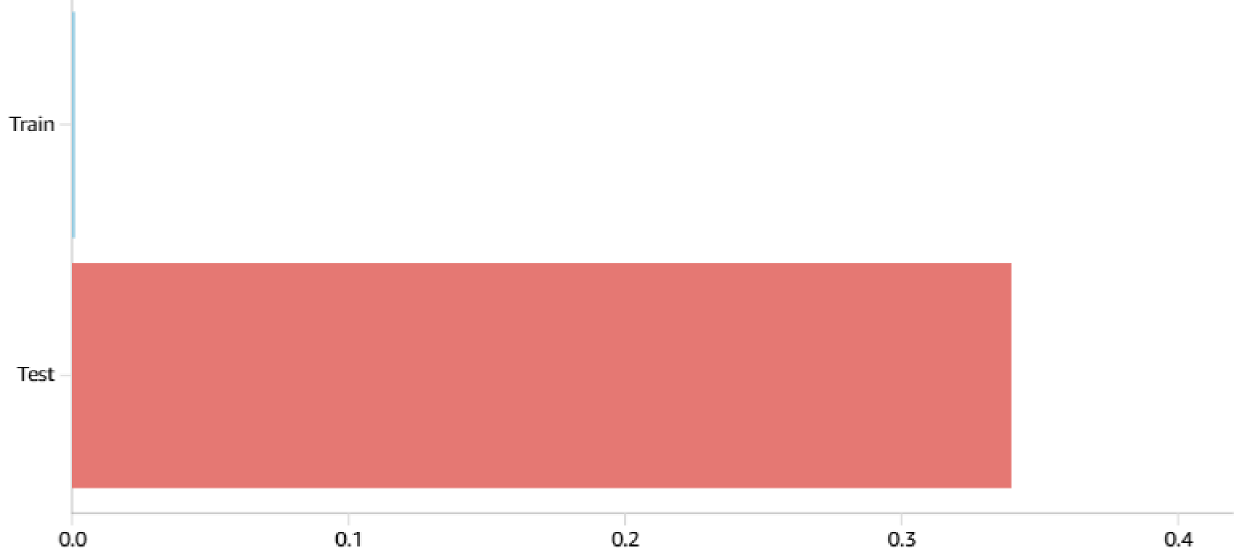
- Variance, is the error due to excessive complexity in the learning algorithm.
- A model with high variance captures not only the underlying patterns but also the noise in the training data.
- This leads to poor generalization to unseen data.

Example of High Variance:

Imagine training a decision tree with a very deep structure on a dataset of handwritten digits. While the tree can fit the training data perfectly, it might perform poorly on new, unseen digits because it has essentially memorized the training examples, including their individual quirks.



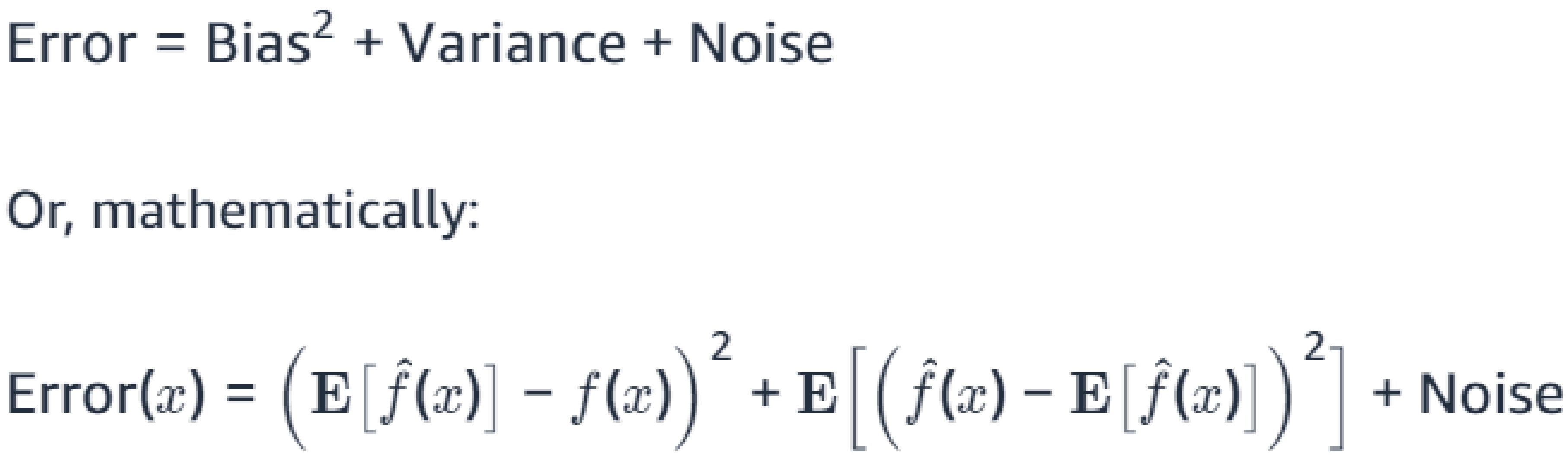
Mean-Squared Error



Test Error Decomposition

Test error can come as a result of both under- and over- fitting of the data, but how do the two relate to each other?

In the general case, mean-squared error can be decomposed into three components: error due to bias, error due to variance, and error due to noise.

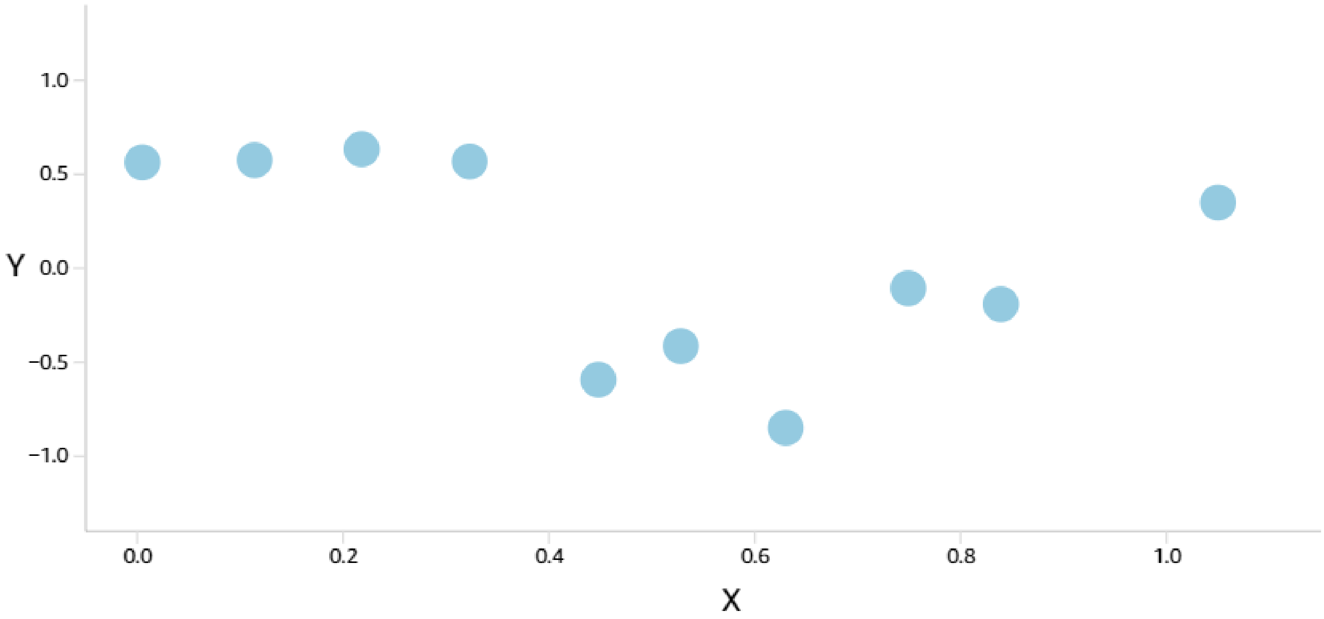


$$\text{Error} = \text{Bias}^2 + \text{Variance} + \text{Noise}$$

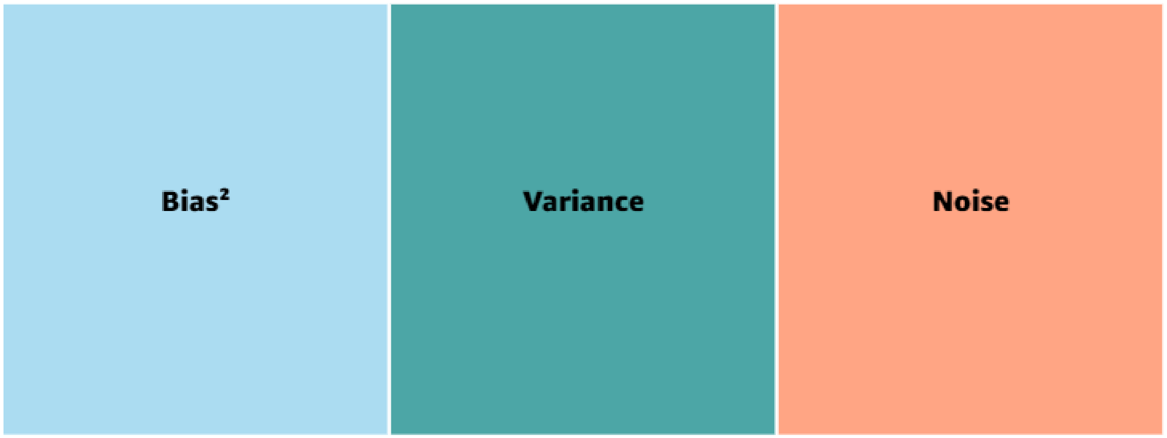
Or, mathematically:

$$\text{Error}(x) = \left(\mathbf{E}\left[\hat{f}(x)\right] - f(x)\right)^2 + \mathbf{E}\left[\left(\hat{f}(x) - \mathbf{E}\left[\hat{f}(x)\right]\right)^2\right] + \text{Noise}$$

We can't do much about the irreducible Noise term, but we can make use of the relationship between both bias and variance to obtain better predictions.



Test Error Decomposition



Bias^2

Variance

Noise

Bias

Bias represents the difference between the average prediction and the true value:

$$\text{Bias}^2(x) = \left(\mathbf{E}\left[\hat{f}(x)\right] - f(x)\right)^2$$

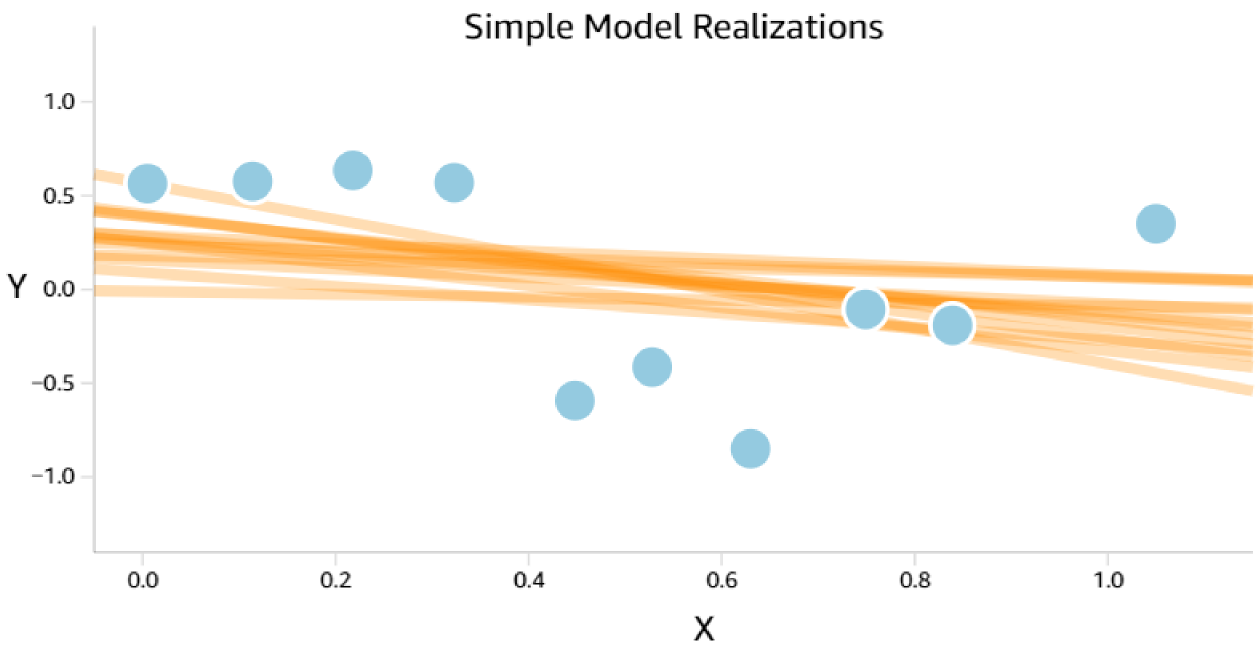
The term $E[\hat{f}(x)]$ is a tricky one.

It refers to the average prediction after the model has been trained over several independent datasets.

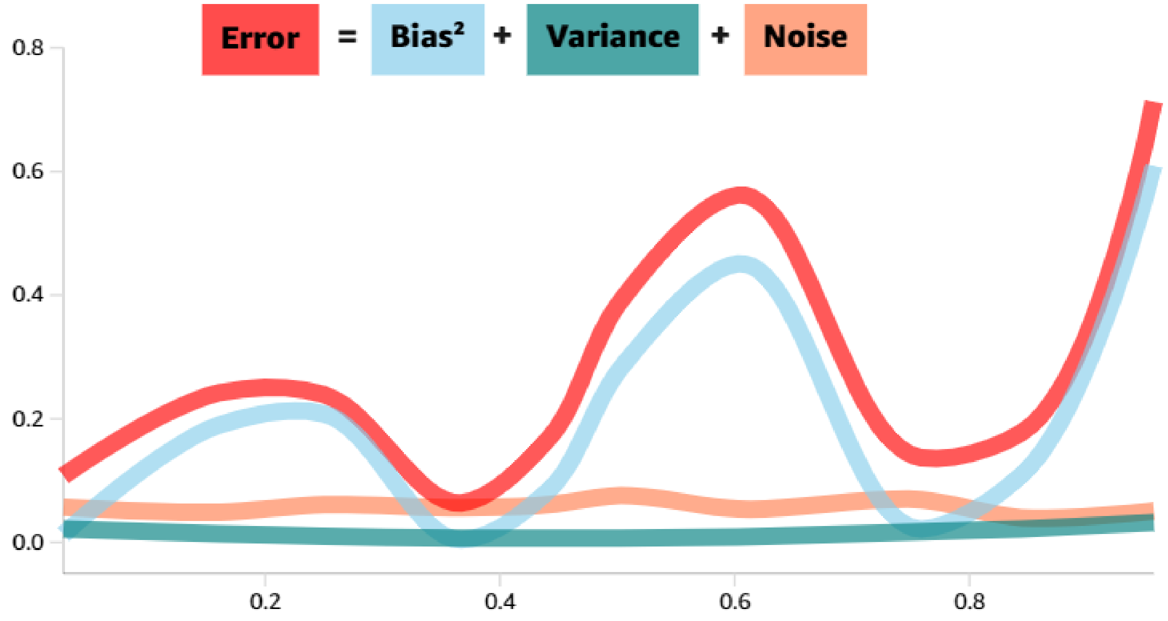
We can think of the bias as measuring a *systematic* error in prediction.

These different model realizations are shown in the below chart, while the error decomposition (for each point of data) is shown in the bottom chart.

For underfit (low-complexity) models, the majority of error comes from bias.



Simple Model Realizations



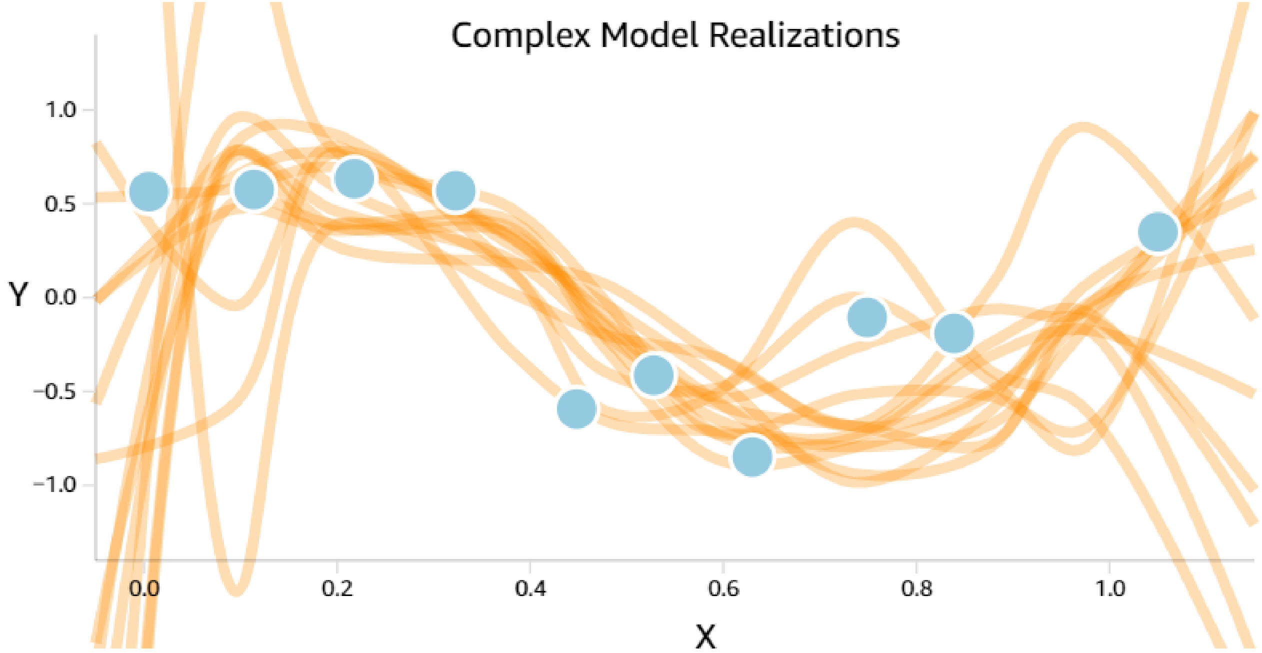
Variance

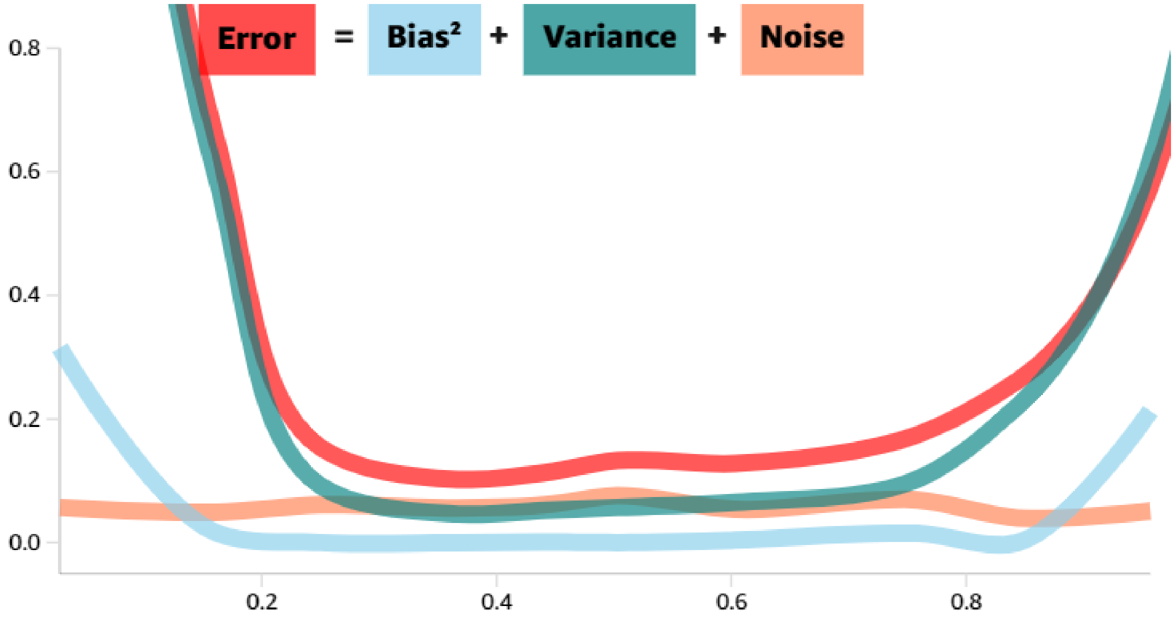
As with bias, the notion of variance also relates to different realizations of our model. Specifically, variance measures how much, on average, predictions vary for a given data point:

$$\text{Variance}(x) = \mathbf{E}\left[\left(\hat{f}(x) - \mathbf{E}\left[\hat{f}(x)\right]\right)^2\right]$$

In the bottom plot, predictions from overfit (high-complexity) models show a lot more error from variance than from bias.

It’s easy to imagine that any unseen data points will be predicted with high error.


Complex Model Realizations
Y
1.0
0.5
0.0
−0.5
−1.0
0.0
0.2
0.4
0.6
0.8
1.0
X


Error
=
Bias²
+
Variance
+
Noise
0.8
0.6
0.4
0.2
0.0
0.2
0.4
0.6
0.8

Identifying Underfitting and Overfitting

Underfitting

- Underfitting occurs when a model is too simplistic to capture the underlying patterns in the data.

- We can identify underfitting by analyzing the model’s performance on both the training and test datasets:

•Training Set: A model with high bias will struggle to fit the training data, resulting in low training accuracy (e.g., 60%).

•Test Set: The same model will also perform poorly on the test set, leading to low test accuracy (e.g., 65%).

In summary, underfitting results in a model that fails to grasp even the fundamental relationships in the data.

Overfitting

Overfitting happens when a model is excessively complex, capturing not just the patterns but also the noise in the training data.

Identifying overfitting is a bit more nuanced:

•Training Set: A model with high variance can fit the training data exceedingly well, achieving high training accuracy (e.g., 95%).

•Test Set: However, this model will perform significantly worse on the test set, exhibiting lower test accuracy (e.g., 75%).
In essence, an overfit model “memorizes” the training data but struggles to generalize to new, unseen data.

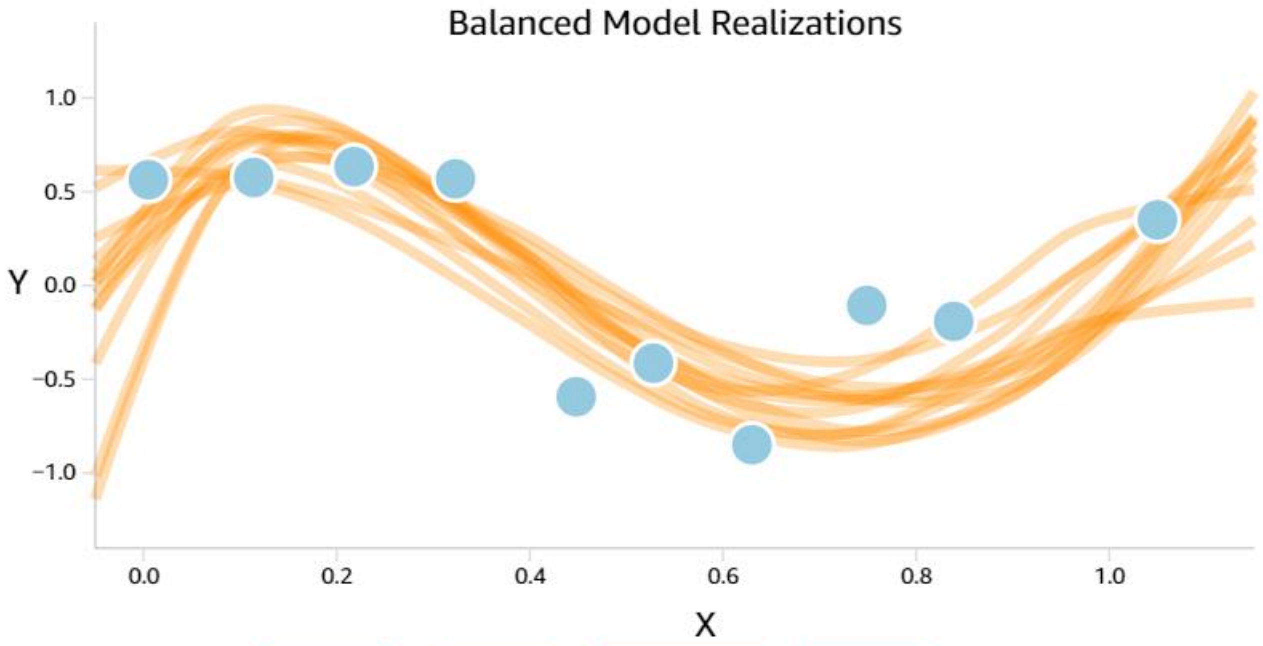
Finding A Balance

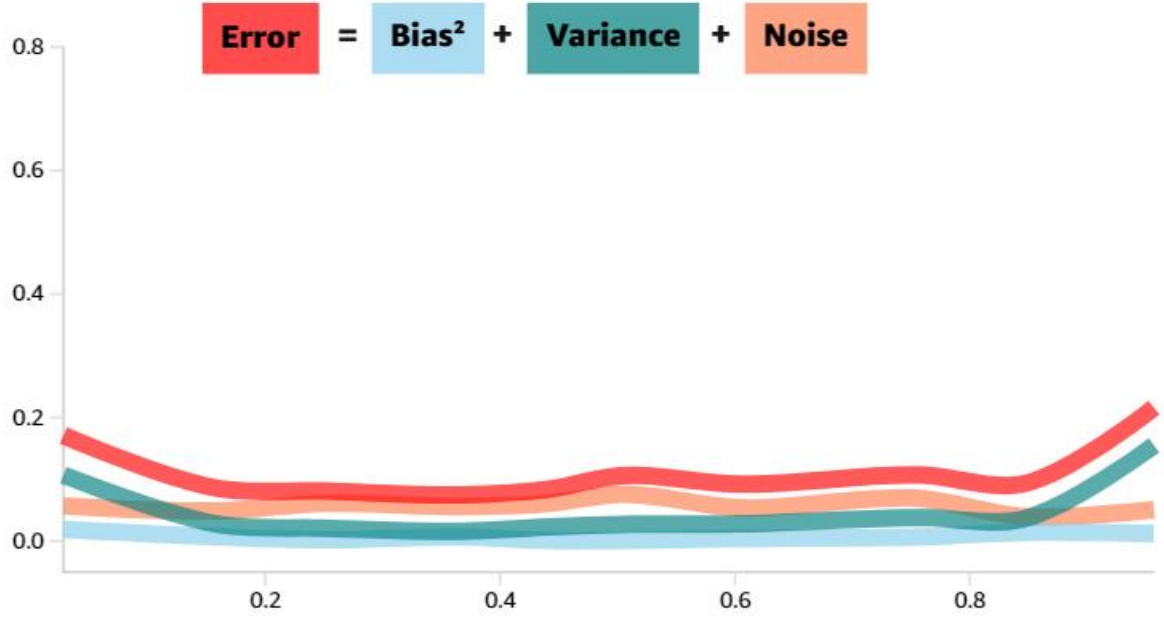
To obtain the best results, we should work to find a happy medium between a model that is so basic it fails to learn meaningful patterns in our data, and one that is so complex it fails to generalize to unseen data .

In other words, we don’t want an underfit model, but we don’t want an overfit model either. We want something in between - something with enough complexity to learn learn the generalizable patterns in our data.

By trading some bias for variance (i.e. increasing the complexity of our model), and without going overboard, we can find a balanced model for our dataset.

At different levels of complexity, a sample of model realizations along side their corresponding prediction error decompositions.


Balanced Model Realizations
Y
1.0
0.5
0.0
−0.5
−1.0
0.0
0.2
0.4
0.6
0.8
1.0
X

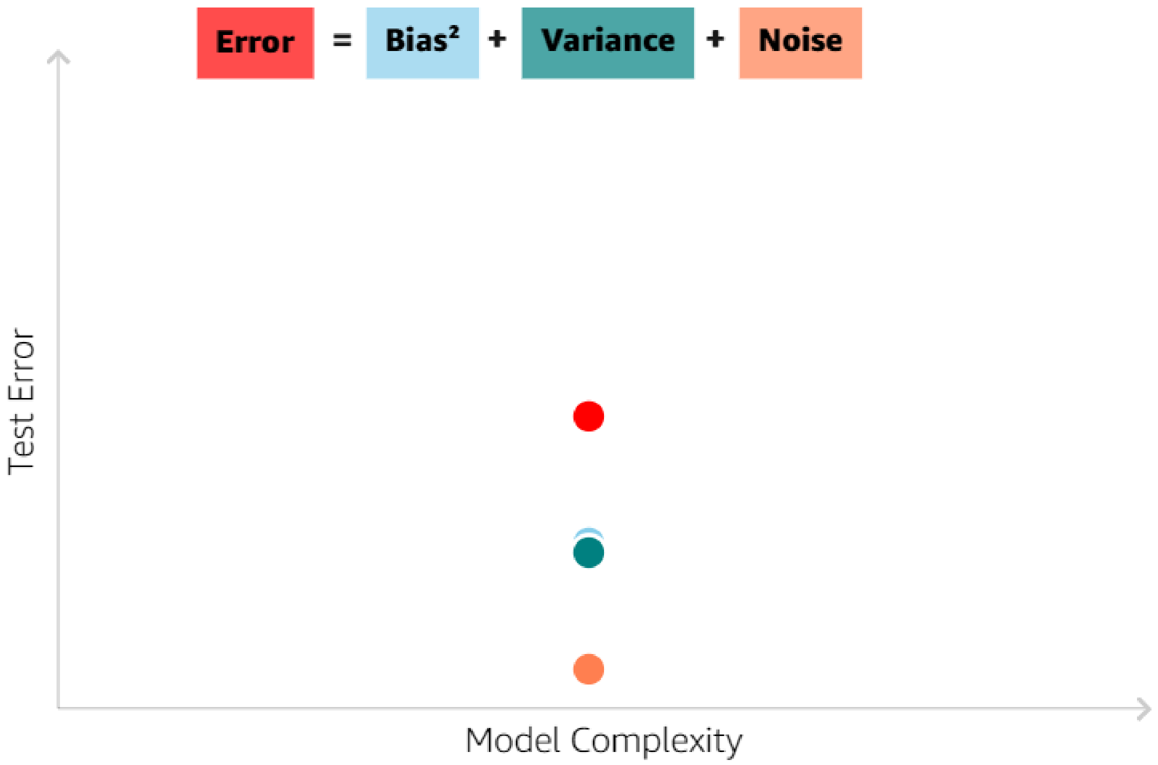

Error = Bias² + Variance + Noise
0.8
0.6
0.4
0.2
0.0
0.2
0.4
0.6
0.8

Across Complexities

- Focus to the error decompositions across model complexities.

  For each level of complexity, aggregate the error decomposition across all data-points, and plot the aggregate errors at their level of complexity.

  This aggregation applied to the balanced model (i.e. the middle level of complexity) is shown in the chart at right side.



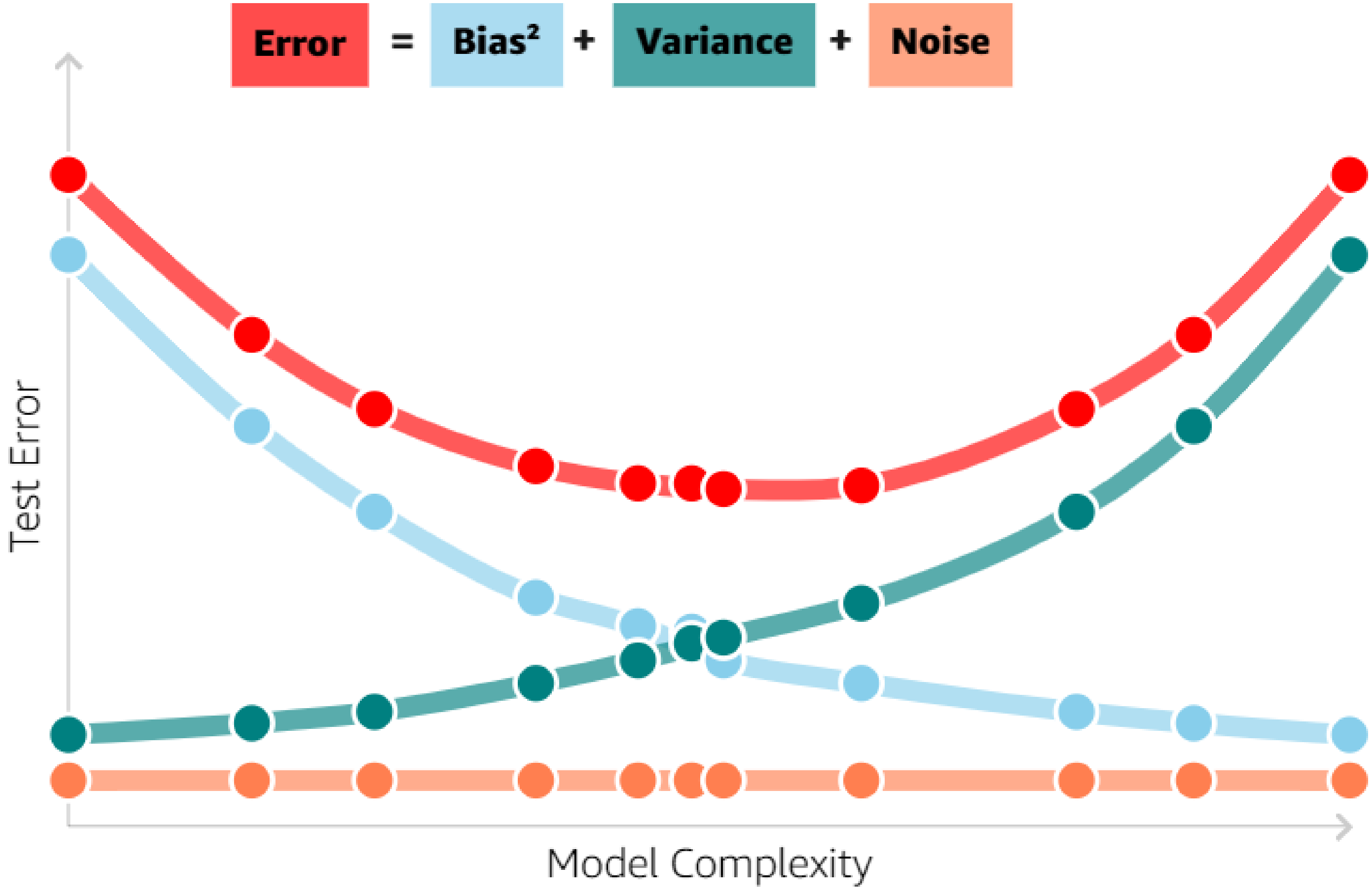
The Bias Variance Trade-off

- Repeating this (discussed in previous slide) aggregation across our range of model complexities, we can see the relationship between bias and variance in prediction errors manifests itself as a U-shaped curve detailing the trade off between bias and variance.

  When a model is too simple (i.e. small values along the x-axis), it ignores useful information, and the error is composed mostly of that from bias.

  When a model is too complex (i.e. large values along the x-axis), it memorizes non-general patterns, and the error is composed mostly of that from variance.

  The ideal model aims to minimize both bias and variance. It lays in the sweet spot - not too simple, nor too complex. Achieving such a balance will yield the minimum error.

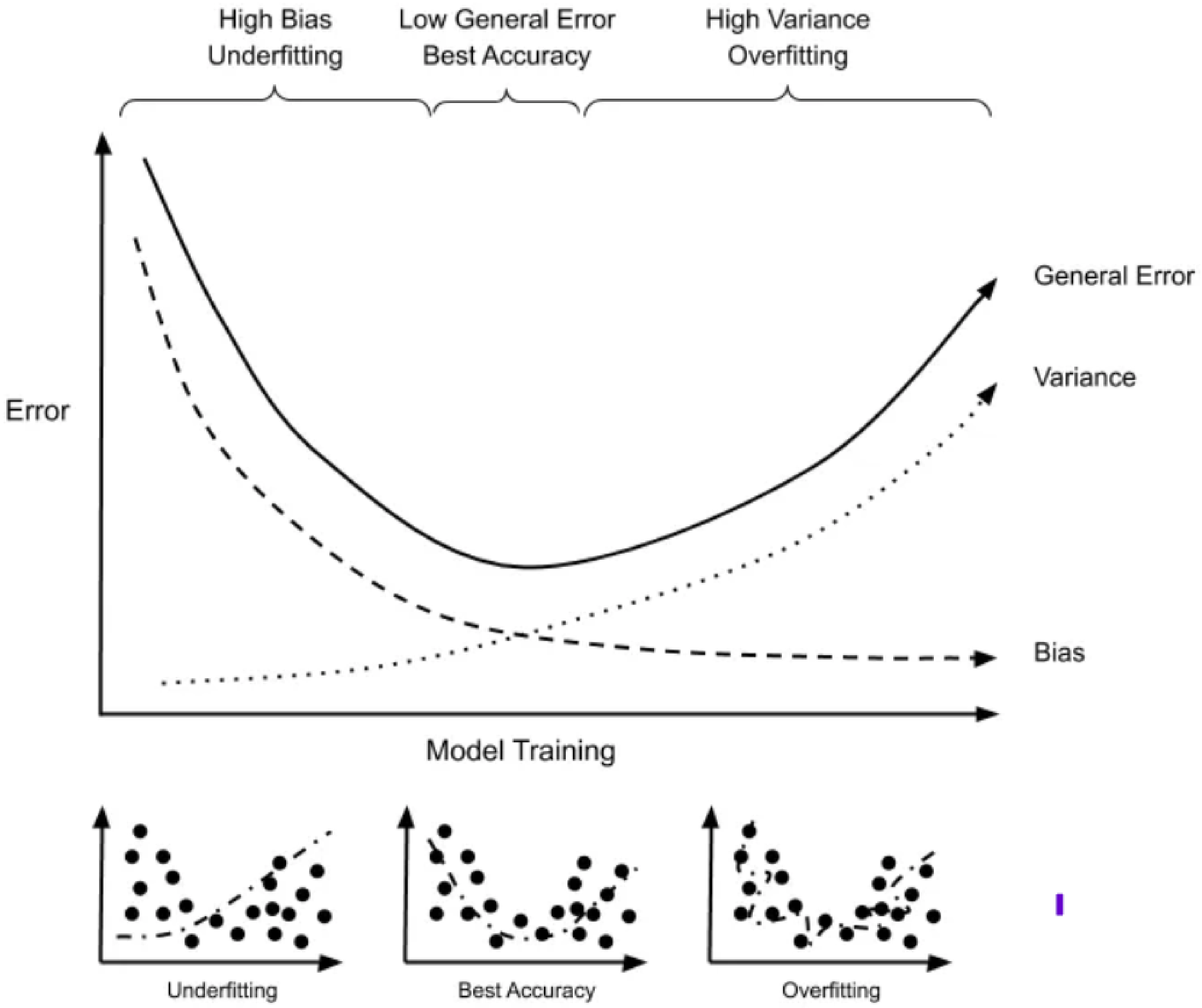

Error
=
Bias²
+
Variance
+
Noise
Test Error
Model Complexity

The Tradeoff: Balancing Bias and Variance

- The bias-variance tradeoff is the delicate equilibrium between underfitting and overfitting.

- The goal is to find the optimal level of complexity that allows a model to generalize effectively to unseen data.

- This tradeoff is often visualized as a curve, known as the **validation error curve**:


High Bias
Underfitting
Low General Error
Best Accuracy
High Variance
Overfitting
Error
General Error
Variance
Bias
Model Training
Underfitting
Best Accuracy
Overfitting

Validation Error Curve:

The validation error curve is a graphical representation that illustrates the relationship between model complexity (or flexibility) and the error on a validation dataset. It typically looks like an inverted U-shape or a curve with two distinct components:

1.Bias (Underfitting) Region: On the left side of the curve, we have the region associated with high bias or underfitting. In this area, the model's complexity is too low to capture the underlying patterns in the data. As a result, both the training and validation errors are high.

2.Variance (Overfitting) Region: On the right side of the curve, we have the region associated with high variance or overfitting. Here, the model's complexity is excessively high, and it starts fitting not only the underlying patterns but also the noise in the training data. In this region, the training error is very low, but the validation error starts to increase significantly because the model fails to generalize to unseen data.

3.Optimal Region (Tradeoff): The point of optimal model complexity lies between the bias and variance regions, often referred to as the "optimal region." This is where the model generalizes well to both the training and validation data, resulting in the lowest validation error.

Solutions to Address Bias and Variance

To strike the right balance and address bias and variance issues, consider the following solutions:

1. **Regularization:** Regularization techniques like L1 (Lasso) and L2 (Ridge) can help mitigate overfitting. These methods add penalty terms to the model’s cost function, discouraging it from becoming overly complex.
2. **Feature Engineering:** Thoughtful feature selection and engineering can reduce both bias and variance. By including relevant features and excluding noisy ones, you can improve model performance.
3. **Cross-Validation:** Utilize cross-validation to assess your model’s performance on different subsets of the data. This helps you gauge how well your model generalizes across various data splits, providing valuable insights into bias and variance.
4. **Ensemble Methods:** Ensemble techniques such as Random Forests and Gradient Boosting combine multiple models to achieve better performance. They can effectively reduce overfitting while improving predictive accuracy.
5. **Collect More Data:** If your model suffers from high bias (underfitting), acquiring more data can help it capture more complex patterns. Additional data can be especially beneficial when dealing with deep neural networks.