



# Model Evaluation



## Companion Book

Model Evaluation is a fundamental topic of understanding your model's performance!

Review Chapter 2 of **Introduction to Statistical Learning** for a more in depth look!



# Model Evaluation

- We'll discuss the following topics
  - Train Test Splits
  - Holdout Sets
  - Parameter Grids
  - Scala and Spark for Model Evaluation
  - Bias Variance Trade-Off
  - Documentation Exploration
  - Code through some Examples



# Train Test Splits

- We've previously talked about Train Test Splits, but let's review the concept.
- You will always train a Machine Learning Algorithm on some data, but afterwards you will want some measure of how well it performed.
- Each main Machine Learning Task has different metrics for evaluation



# Train Test Splits

- Regression
  - $R^2$
  - RMSE
- Classification
  - Precision
  - Recall
- Clustering
  - Within Sum of Squares Error



# Train Test Splits

- While you could get these measurements using the same data you trained your model on, that is not a good idea.
- Your model has already seen this data meaning it is not a good choice for evaluating your model's performance
- You should get these metrics off test data, which your model has not seen yet.
- This is known as a train-test split.



# Holdout Data

- An expansion of this idea is the holdout data set.
- This is separate from the training and test sets.
- In this process you use the training data to fit your model, you use the test set to evaluate and adjust your model.
- You can use the test set over and over again.
- Finally, before deploying your model, you check it against the holdout to get some final metrics on performance.



# Parameter Grids

- As we've seen, we can add optional parameters to Machine Learning Algorithms.
- Many times it is difficult to know what are good values for these parameters.
- Spark makes it possible to set up a grid of parameters to train across.
- You create multiple models, train them across the grid, and Spark reports back which model performed best.





# Spark and Scala for Model Evaluation

- Spark makes all of these processes generally easy with the use of 3 object types:
  - Evaluators
  - ParamGridBuilders
  - TrainValidationSplit
- Later on in this section we will explore how to use these object types to implement the ideas discussed here.



# Spark and Scala for Model Evaluation

- An important aspect to understanding all of this is the Bias-Variance Trade-Off.
- We won't directly explore this with Spark and Scala because it pertains more to theory than Data Engineering.
- However let's take the time now to at least understand the concept so we can have full context for this section of the course.



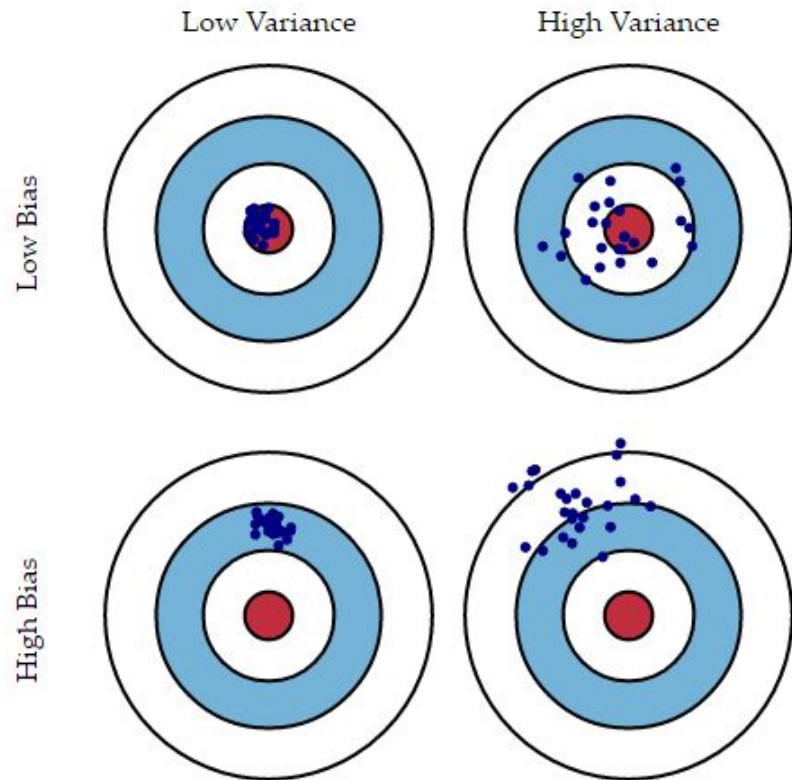
# Bias Variance Trade-Off

- The bias-variance trade-off is the point where we are adding just noise by adding model complexity (flexibility).
- The training error goes down as it has to, but the test error is starting to go up.
- The model after the bias trade-off begins to overfit.



# Bias Variance Trade-Off

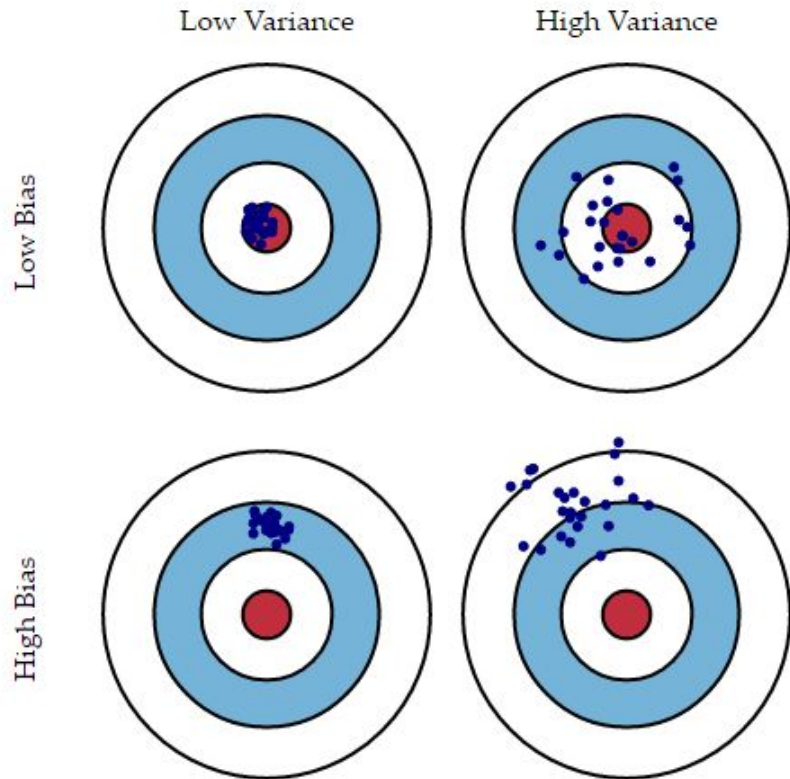
- Imagine that the center of the target is a model that perfectly predicts the correct values.
- As we move away from the bulls-eye, our predictions get worse and worse.





# Bias Variance Trade-Off

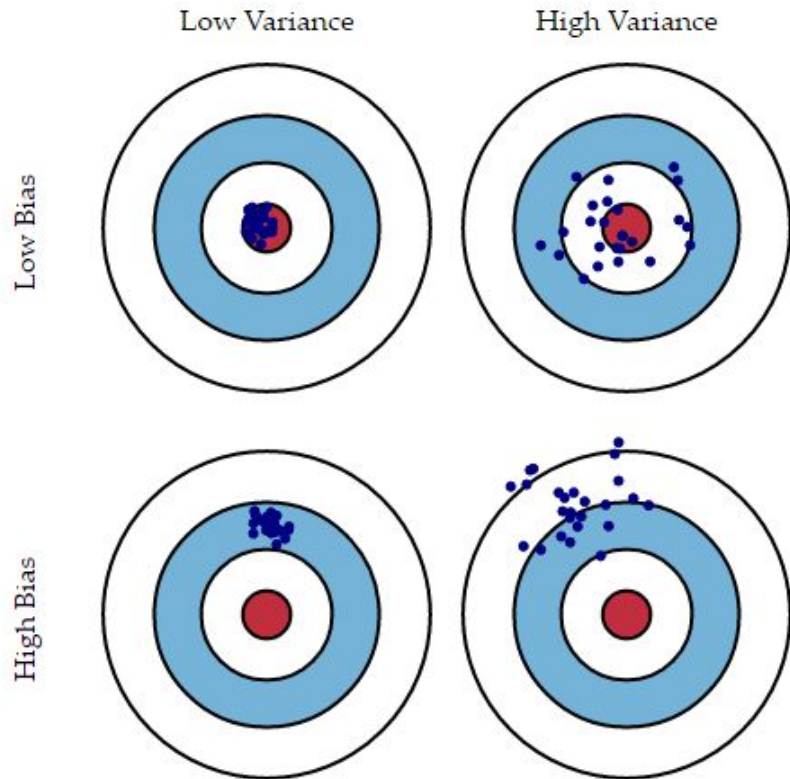
- Imagine we can repeat our entire model building process to get a number of separate hits on the target.
- Each hit represents an individual realization of our model, given the chance variability in the training data we gather.





# Bias Variance Trade-Off

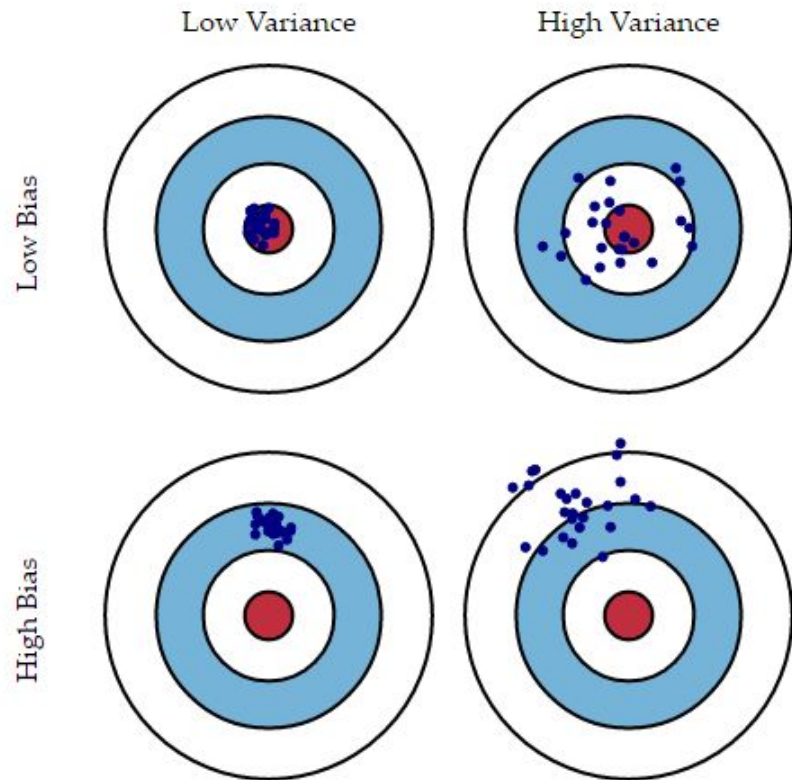
- Sometimes we will get a good distribution of training data so we predict very well and we are close to the bulls-eye, while sometimes our training data might be full of outliers or non-standard values resulting in poorer predictions.





# Bias Variance Trade-Off

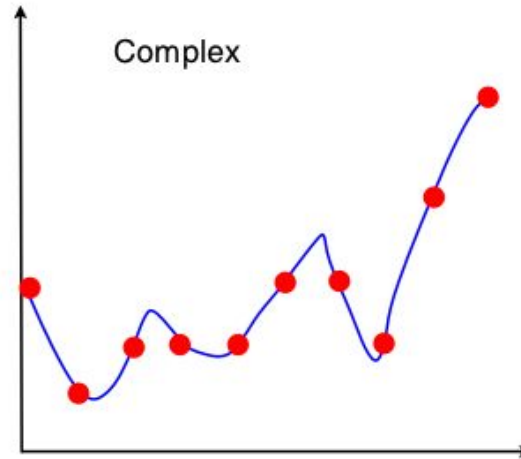
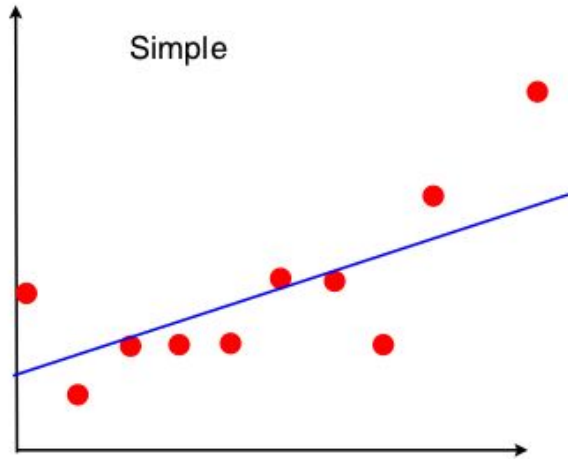
- These different realizations result in a scatter of hits on the target.





# Bias Variance Trade-Off

- A common temptation for beginners is to continually add complexity to a model until it fits the training set very well.





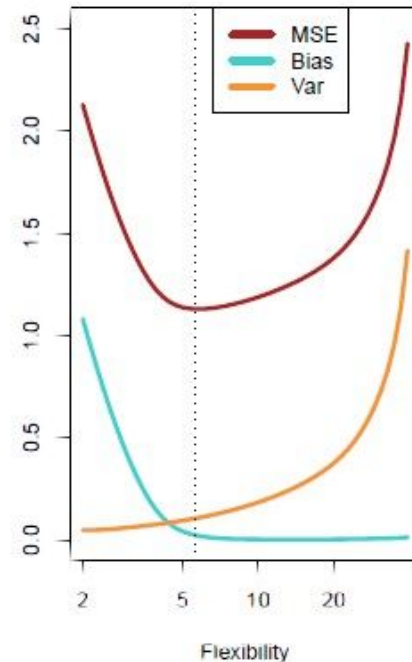
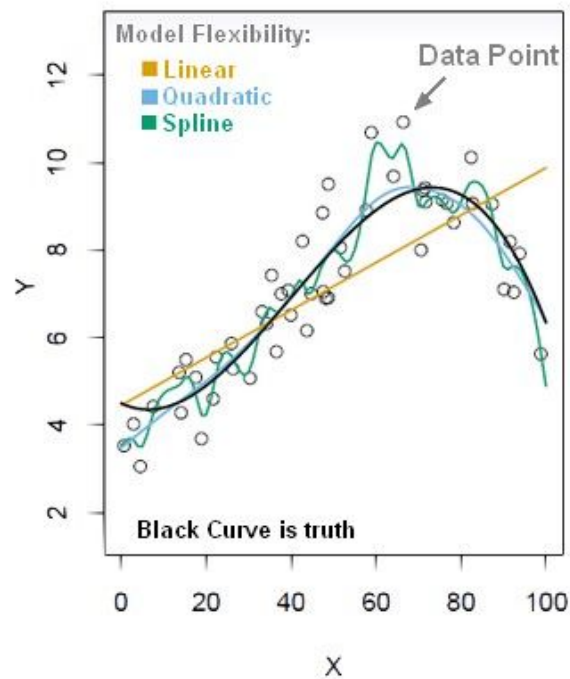


# Bias Variance Trade-Off

- Doing this can cause a model to overfit to your training data and cause large errors on new data, such as the test set.
- Let's take a look at an example model on how we can see overfitting occur from a error standpoint using test data!
- We'll use a black curve with some “noise” points off of it to represent the True shape the data follows.

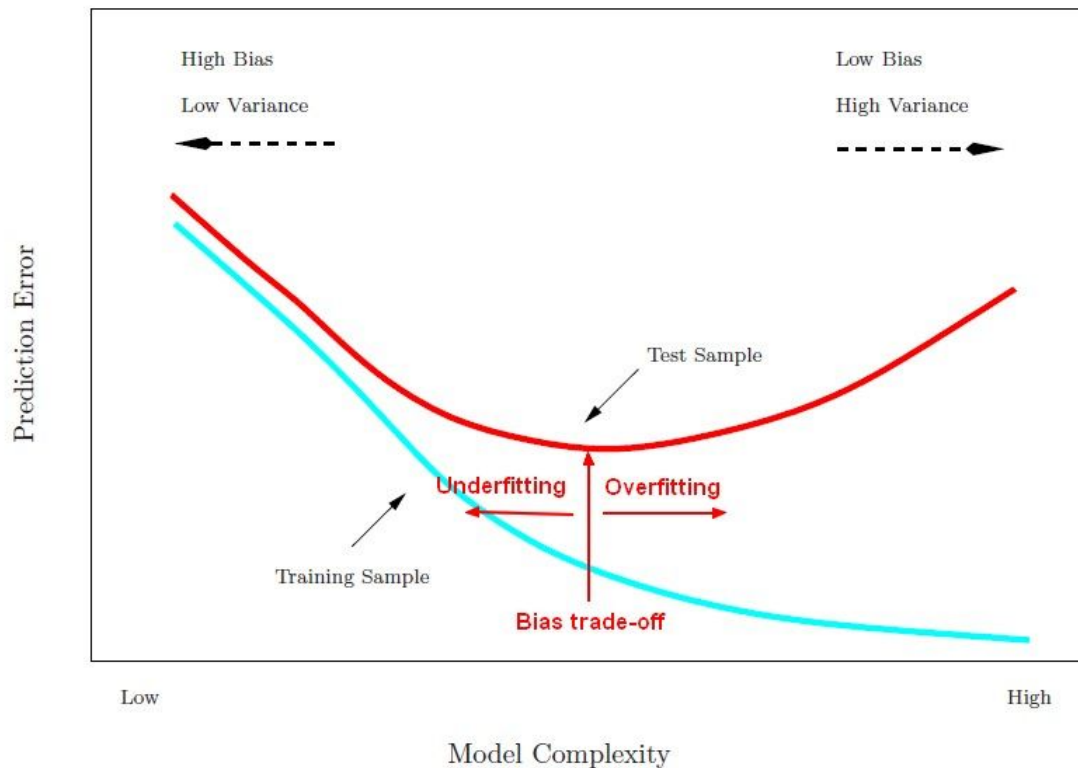


# Bias Variance Trade-Off





# Bias Variance Trade-Off





**Let's continue by going  
through some examples and  
exploring the documentation!**

