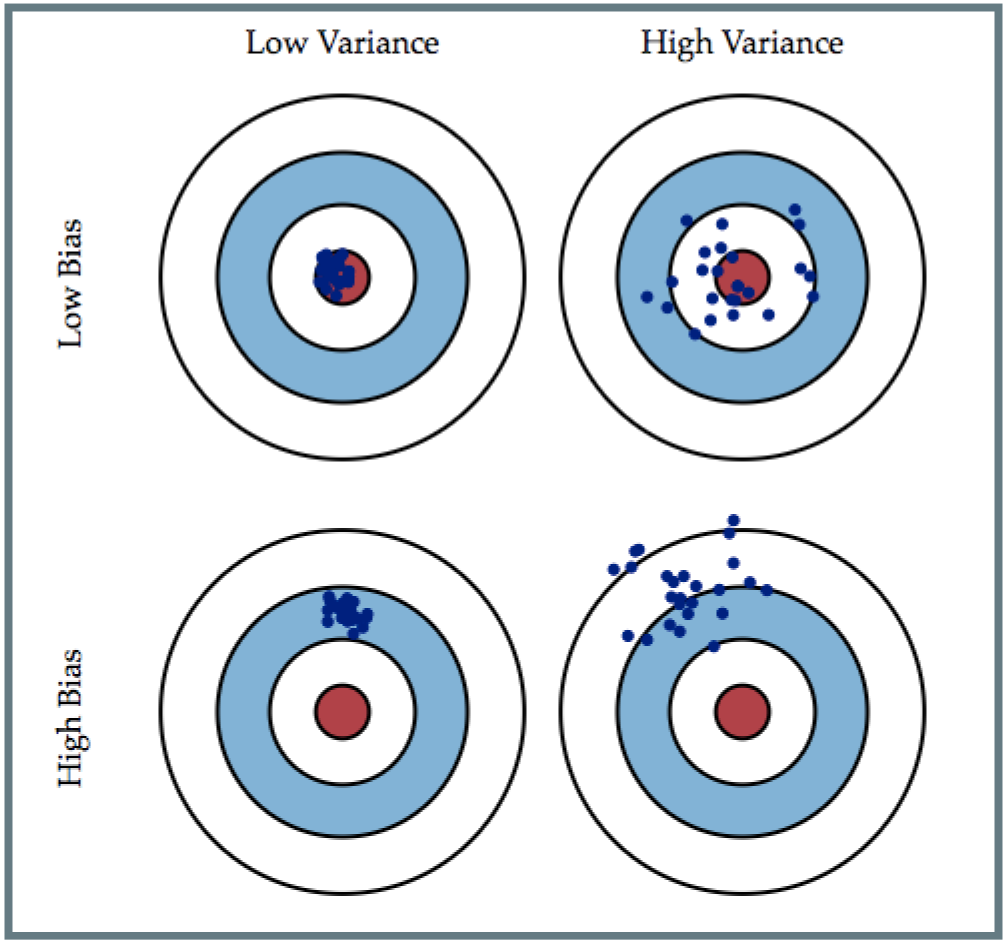
UNDERSTANDING THE BIAS-VARIANCE TRADEOFF

BIAS AND VARIANCE IN MACHINE LEARNING

BIAS AND VARIANCE

- **Bias**: The error due to bias is taken as the difference between the expected (or average) prediction of our model and the correct value which we are trying to predict.
- **Variance**: The error due to variance is taken as the variability of a model prediction for a given data point.

BIAS AND VARIANCE



BIAS AND VARIANCE

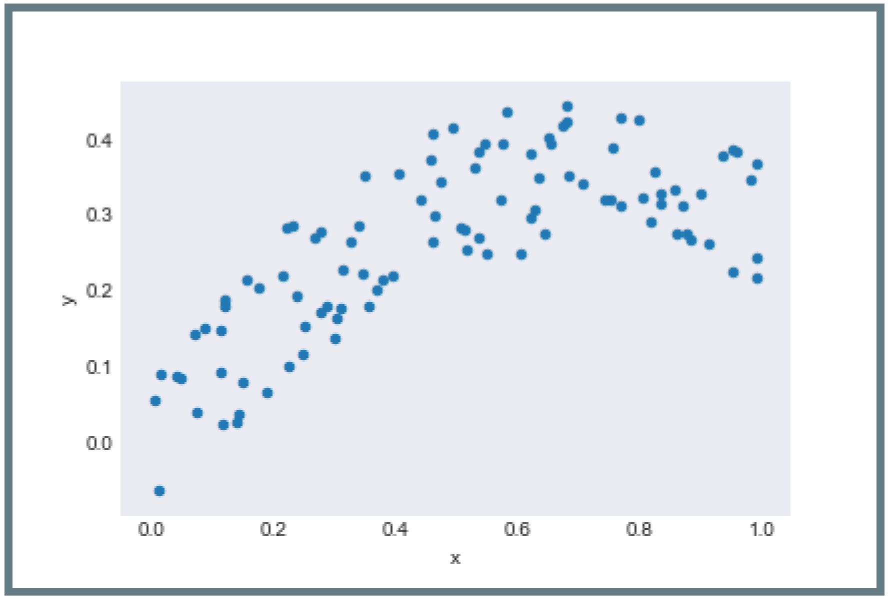
- High bias means our model is not getting close to the truth
- Low bias means our model is (on average) getting close to the truth
- High variance means that the predictions from our model vary a lot across repeated samples
- Low variance means that the predictions from our model don’t vary a lot across repeated samples

AN ILLUSTRATIVE EXAMPLE

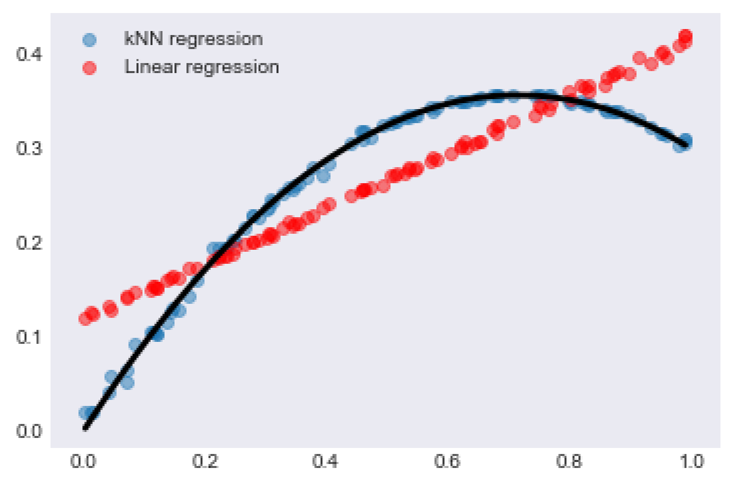
We simulated data with two variables x and y:

$$y = x - 0.8x^2$$

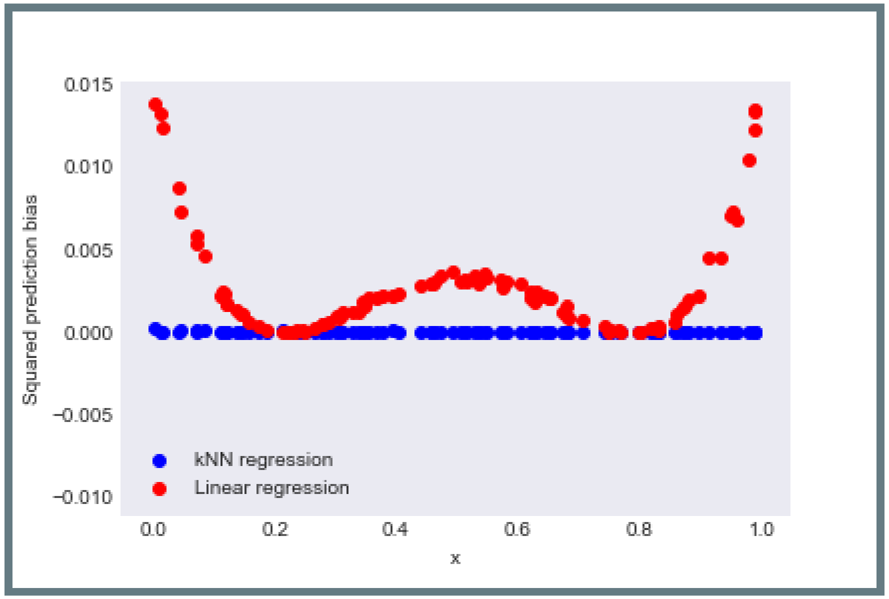
We added random noise to the y-values.



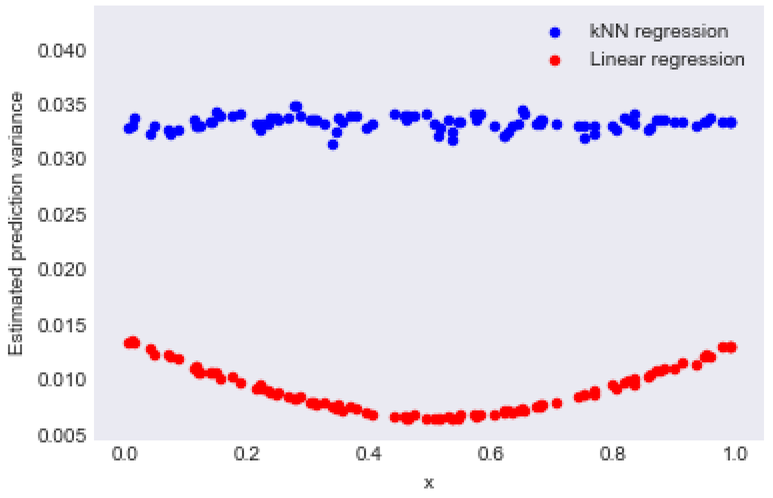
AN ILLUSTRATIVE EXAMPLE



AN ILLUSTRATIVE EXAMPLE



AN ILLUSTRATIVE EXAMPLE



BIAS-VARIANCE TRADEOFF IN TREES

We know that deep trees tend to overfit (ergo, high variance)

Shallow trees tend to be biased

So you grow a tree and then *prune it* to create the Goldilocks tree (just right)

This can be done by cross-validation

PENALIZED REGRESSION AND TREE PRUNING

- Penalized regression takes a loss function and adds a penalty term for complexity
- The loss function has to overcome the penalty of getting more complex
- The optimum number of parameters is often chosen by cross-validation
- Lasso, Ridge Regression. Elastic Net
- The tree pruning step can also be viewed as a penalized optimization step

THE BOOTSTRAP

BASIC IDEA

- We would ideally like to get the sampling distribution of a statistic
- This would require sampling from the data generative process (unknown)
- We still really want this
- Efron and others figured out that if you resampled with replacement from the data you have, you get pretty close
- Singh and others established asymptotic convergence of bootstrap distribution to sampling distribution

HOW DOES ONE BOOTSTRAP?

1. Grab your data
2. Sample **with replacement** from the row-index of your data
3. This might lead to some duplication, and some omission
4. Grab the subset of data defined by the sample
5. Compute your statistic of interest
6. Repeat many times (1000s, even)
7. See the distribution of the statistic

HOW DOES ONE BOOTSTRAP?

The issue is how long it takes to compute your statistic on your data

On today’s clusters and cloud, you can parallel process this (embarassingly parallel).

We can make short work of it, and there are shortcuts many take as well

DOES IT REALLY WORK?

Let’s see….

BOOTSTRAP AGGREGATION (BAGGING)

SUMMARIZING OVER BOOTSTRAP SAMPLES

1. We want to train (and test) our model over the bootstrap samples
2. For each bootstrap sample we see which observations weren’t included (OOB sample)
3. We train on the bootstrap sample, test on the OOB sample
4. Our final product is the average of the OOB predictions for each point

LEVERAGING THE POWER OF AVERAGING

- The laws of large numbers tell us that averaging over more things reduces the standard error of the average
- Intuitively this makes sense: we have more information, so we should be more precise (less variable) in our estimates

LEVERAGING THE POWER OF AVERAGING

What does this tell us about the kinds of models that would benefit from bagging?

CREATING A BAGGED DECISION TREE REGRESSOR

RANDOM FORESTS

CREATING A FOREST FROM TREES

We can use bagging to create an ensemble of trees

Brieman (2001) put in a tweak to the bagged trees:

1. Grow the trees to purity
2. At each split, only consider **a random subset of predictors** for the split

START WITH OVERFITTED MODELS? WHAT THE.....

COMING BACK FROM THE BRINK

- Bagging allows the ensemble to reduce overfitting compared to an individual learner
- So we can take low bias high variance models and reduce their variance!!

CHECKING OUT THE DARK CORNERS

- Just doing a straightforward tree can put us in a rut
- Only some powerful predictors are considered
- This only leads to a local optimum
- We need to get away from that rut
- Random selection of predictors at the nodes
- Forced to go looking in the corners
- Hoping to approach a global optimum

LOOKING AT NUMBER OF PREDICTORS

IS IT USEFUL?

- The behavior of models over different levels of max_features seems to depend on the data set
- If there is some strong separation, it can hurt you
- If there is collinearity or good mixing, it can help you