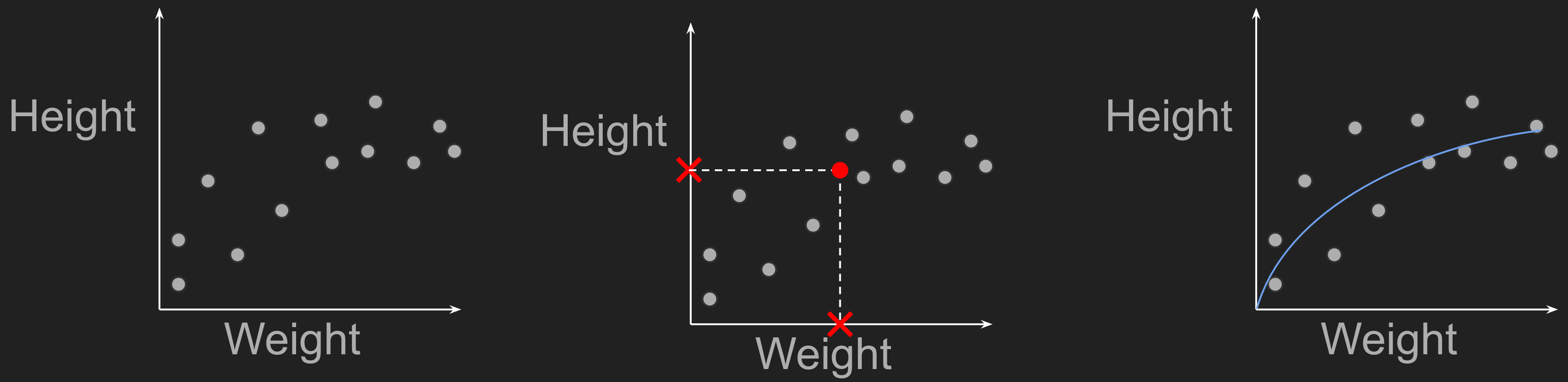
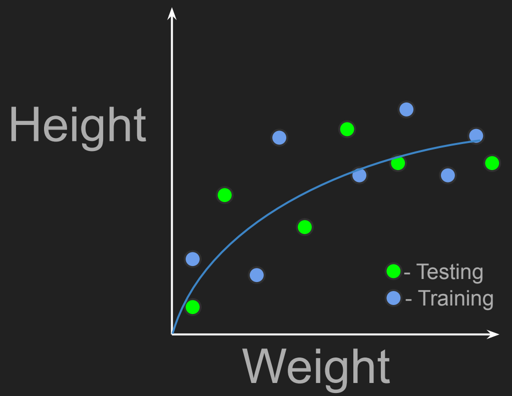
Supervised Machine Learning

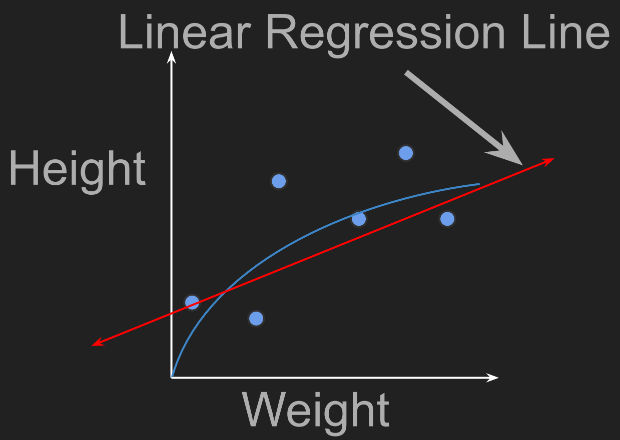
-Mainak Dev

Understanding the Bias Variance tradeoff



Let's suppose we have a graph where we have a relationship between height and weight for mice, low weights mice tend to be short and heavier mice tend to be tall. Given the data we would like to predict the height given a specific weight. Ideally we would know the exact mathematical formula that describes the relationship between weight and height, but since we don't know the formula so we will use two machine learning methods to approximate

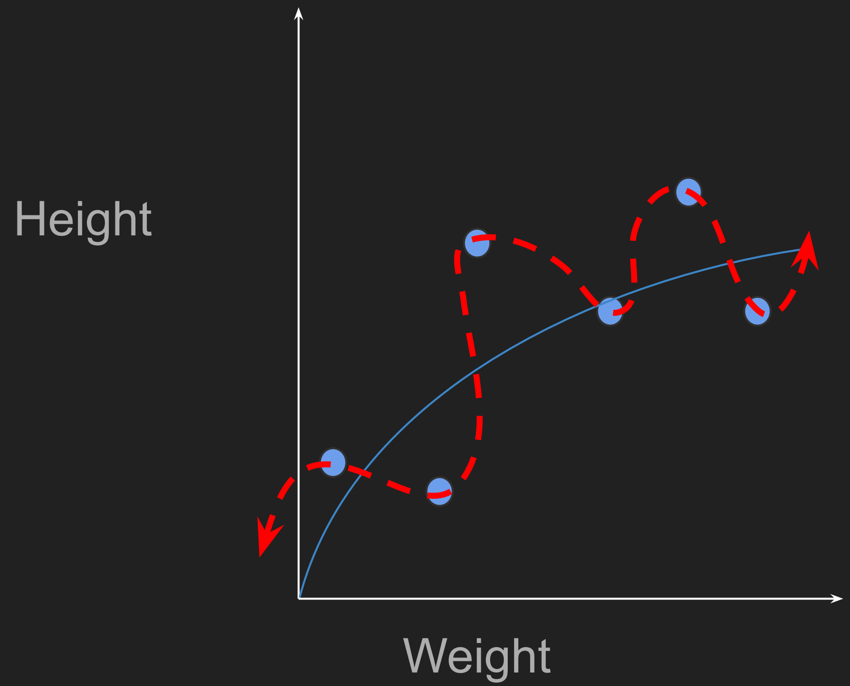




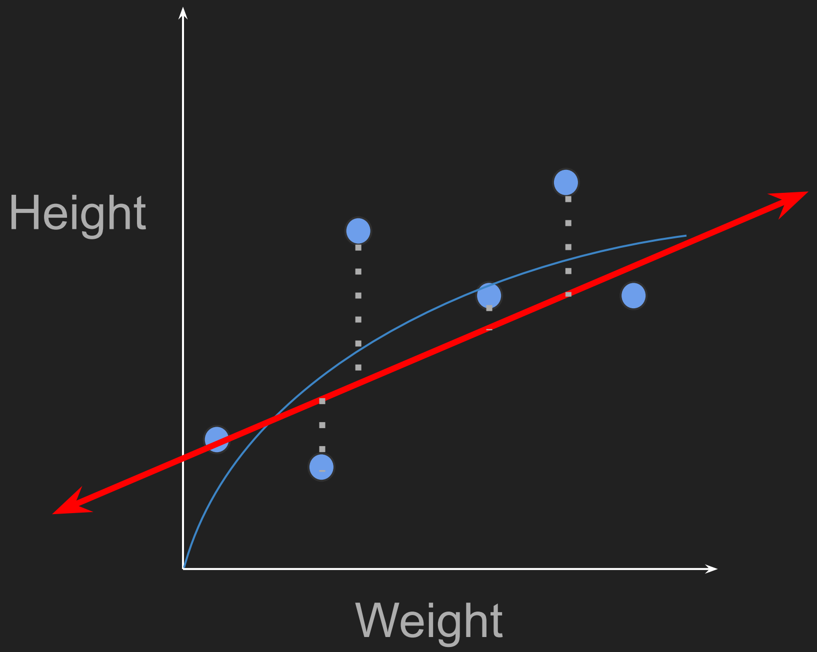
First we will split the data into training and test split

Straight line does not have the flexibility to accurately replicate the arc in the "true" relationship.

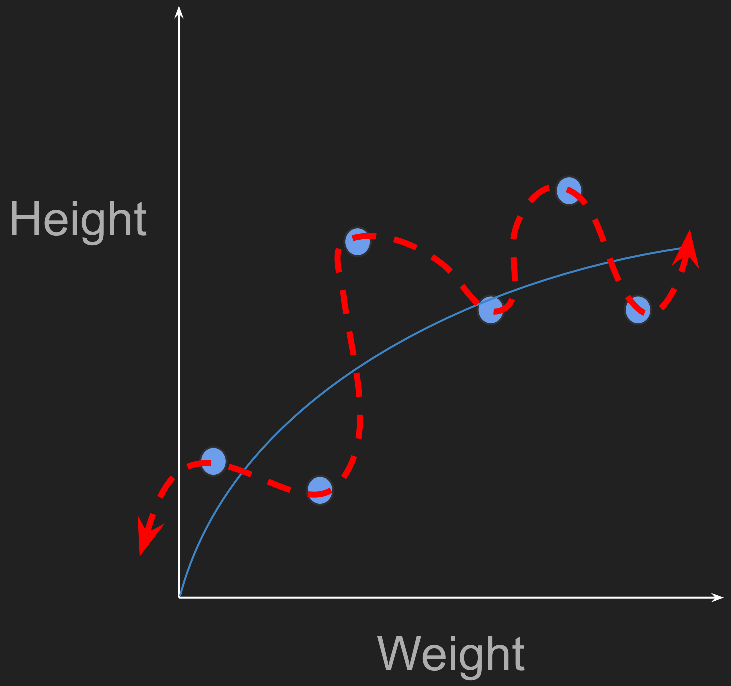
No matter how we try to fit the straight line it will not curve.
"The inability of a machine learning algorithms to capture the true relationship is called bias."



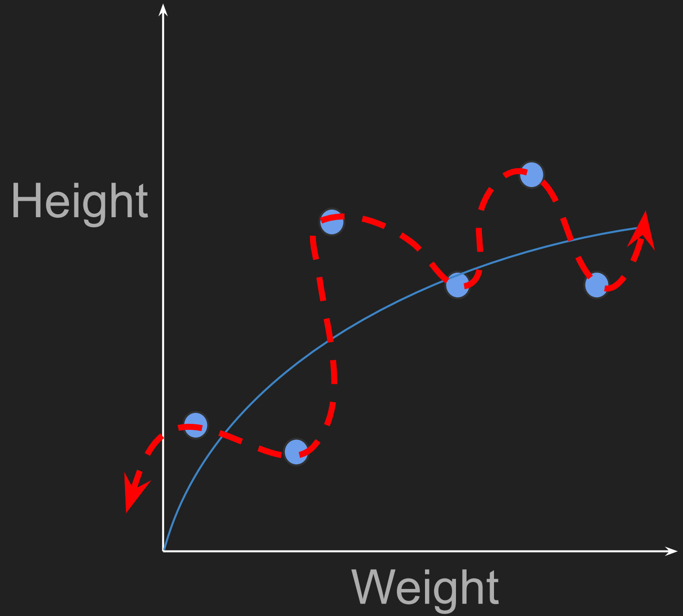
Squiggly line can handle the arc in true relationship thus having very little bias

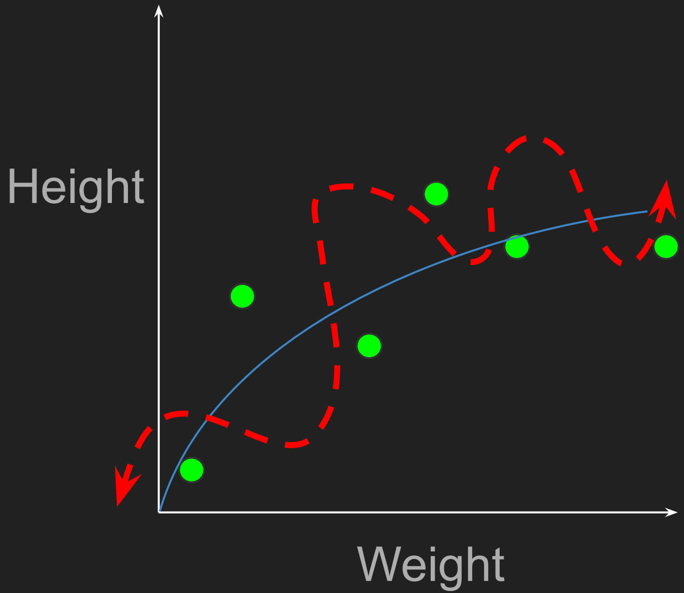


VS



We can compare the Linear Regression line to the the Squiggly using the sum of squares method. But remember we also have a testing set, now let's compare it with the sum of squares of the testing set.



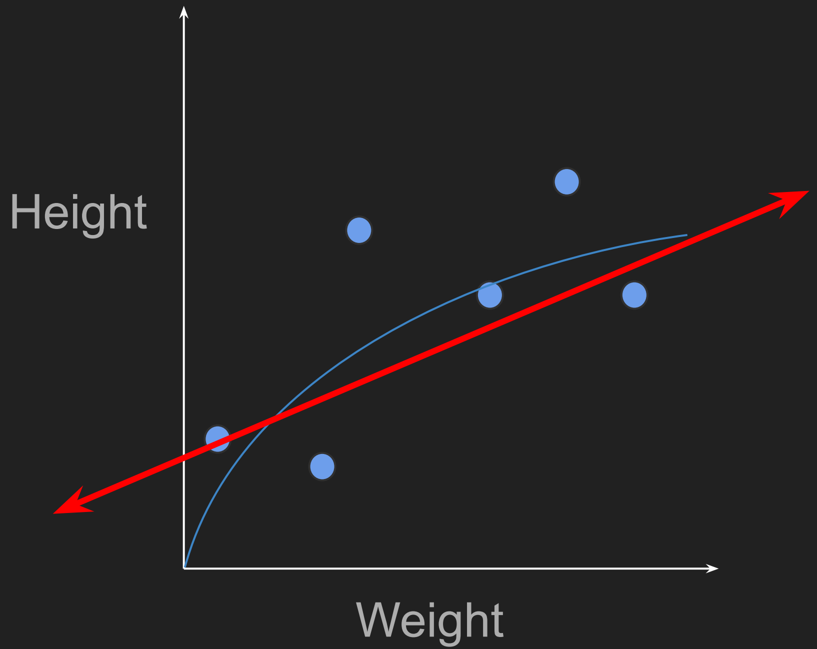


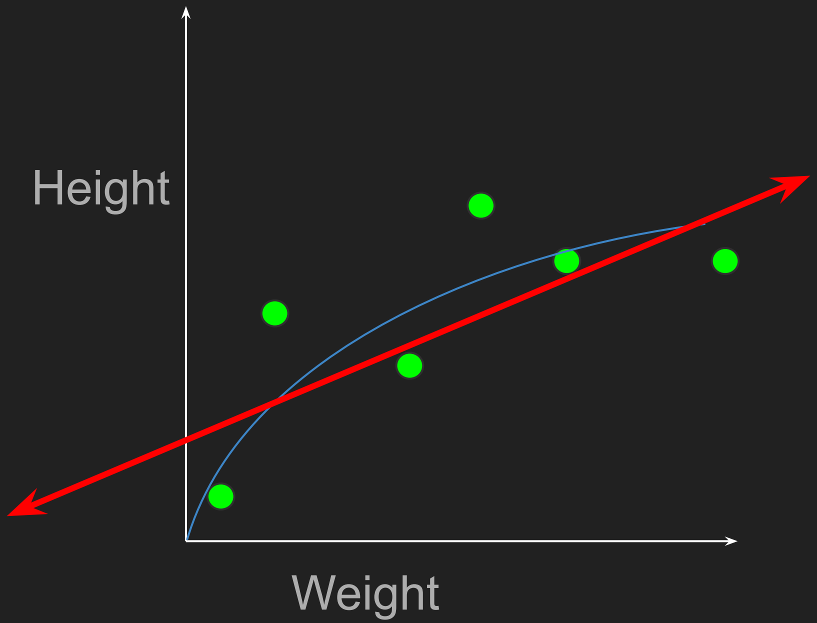
Here, the straight line wins. The squiggly line did a great job in the training set but did terrible on the testing set

The difference in fits between data sets is called **Variance**

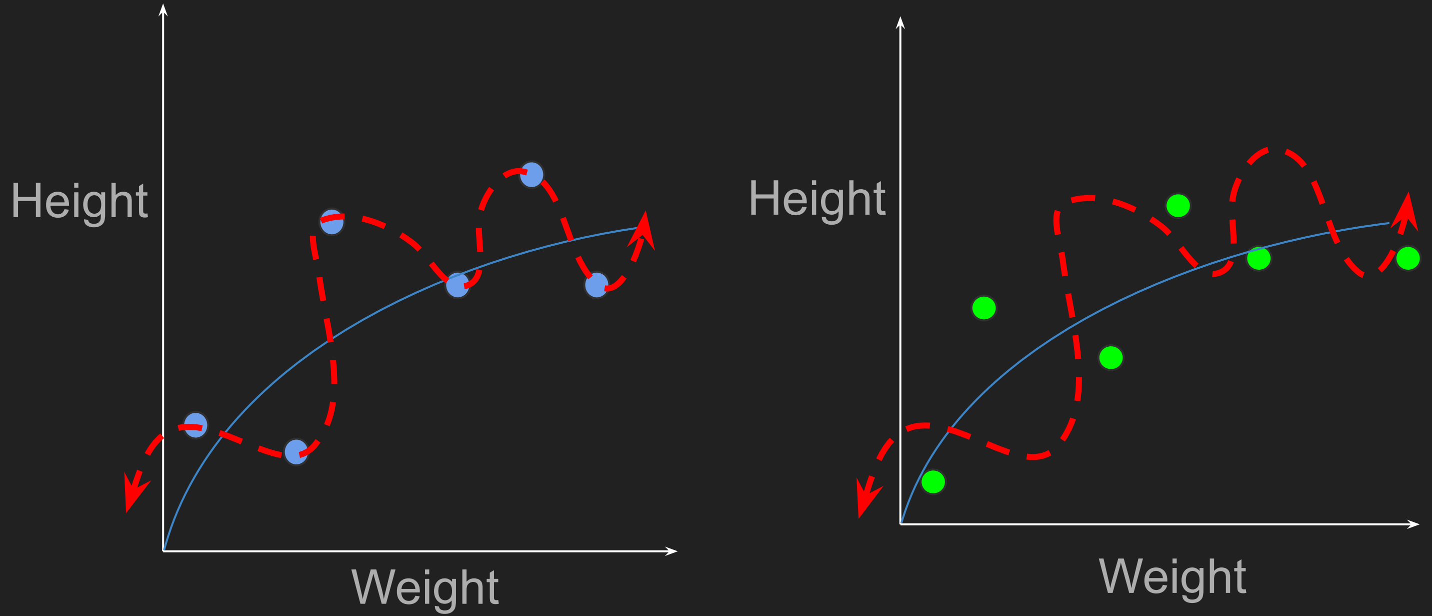
The squiggly line has low bias since it can adapt to the curve in the relationship between weight and height but it has high variability it results in vastly different sums of squares.

Meanings it's difficult to predict how well the squiggly line will perform with future sets.





In contrast the Straight Line has a relatively high bias since it cannot capture the curve in the relationship between weight and height. It has relatively low variance since the Sums of Squares are very similar for different datasets. We can also say that the straight line will give good prediction and not great predictions but they will be consistently good prediction.



Since the squiggly line fits the training set really well, but not the testing set we say the squiggly line is overfit
In Machine learning the ideal algorithm has high bias and low variance by producing consistent predictions across different datasets.
This is done by finding the sweet spot between a simple model and a complex model

Three commonly used methods for finding the sweet spots between simple and complicated models are:

1. Regularization
2. Boosting
3. Bagging

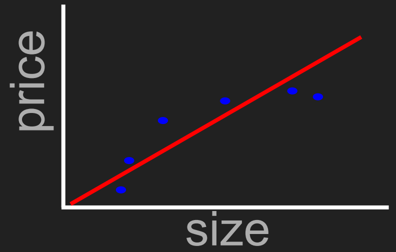
Regularization in Linear Regression

Understanding the Bias Variance tradeoff

Low Bias High Variance: Underfitting

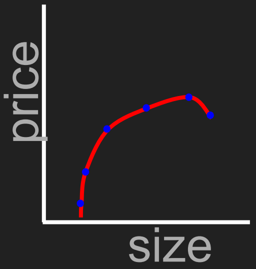
High Bias Low Variance: Overfitting

Example: Linear regression(housing prices)



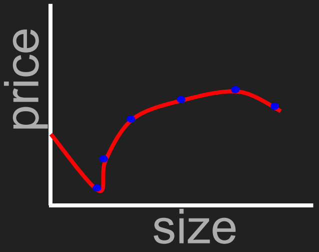
$\theta_0 + \theta_1 x$

Underfitting
Low Bias



$\theta_0 + \theta_1 x + \theta_2 x^2$

Just right



$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Overfitting
High variance

Overfitting: If we have too many features, the learned hypothesis may fit the training set very well($J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^i) - y^i\right)^2 \approx 0$) but fail to generalize to new example(Predice prices on new examples)

Overfitting

Problem with Linear Regression

There are two problems with Linear Regression

Overfitting:

To solve this we can do the following

1. Reduce the number of features
    a. Manually select which features to keep
    b. Model selection algorithm
2. Regularization
    a. Keeps all the features but reduces magnitude of parameters
    b. Works well when we have a lot of features

Underfitting: