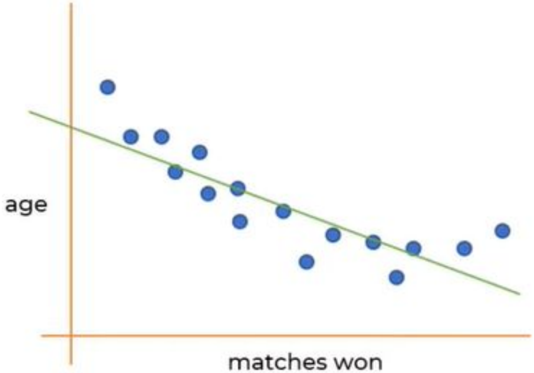
Bias is the simplifying assumptions made by the model to make the target function easier to approximate.

Variance is the amount that the estimate of the target function will change given different training data.

Overfitting: Good performance on the training data, poor generalization to other(test) data.

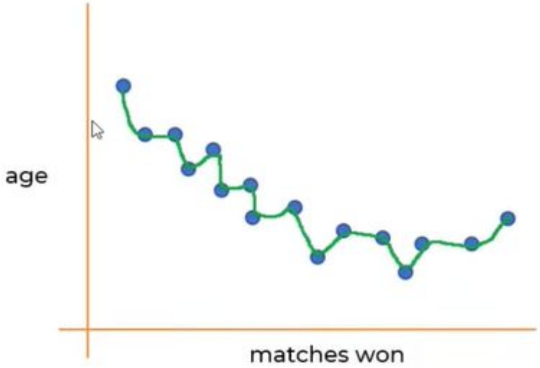
Underfitting: Poor performance on the training data and poor generalization to other(test) data

underfit



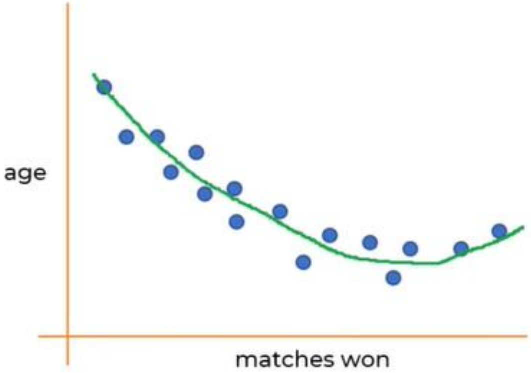
$$\text{match won} = \theta_0 + \theta_1 * age$$

overfit



$$\text{match won} = \theta_0 + \theta_1 * age \quad +\theta_2 * age^2 \\ +\theta_3 * age^3 + \theta_4 * age^4$$

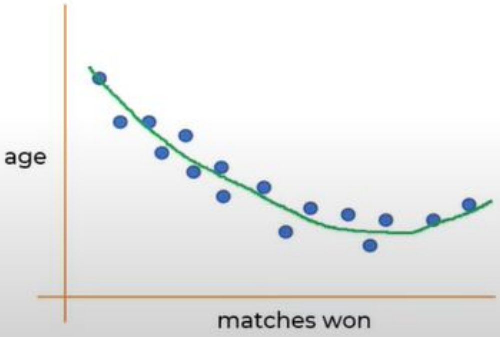
balanced fit



$$\text{match won} = \theta_0 + \theta_1 * age \quad +\theta_2 * age$$

$$\text{match won} = \theta_0 + \theta_1 * age \quad + \theta_2 * age^2 + \theta_3 * age^3 + \theta_4 * age^4$$

Try to make θ_3 and θ_4 almost close to zero



$$\text{match won} = \theta_0 + \theta_1 * age \quad + \theta_2 * age^2$$

Mean Squared Error

$$mse = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_{predicted}\right)^2$$

Mean Squared Error

$$mse = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - h_\theta(x_i)\right)^2$$

$$h_\theta(x_i) = \theta_0 + \theta_1 x_1 + \theta_2 x_2^2 + \theta_3 x_3^3$$

L2 Regularization

$$mse = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - h_\theta(x_i)\right)^2 + \lambda \sum_{i=1}^{n} \theta_i^2$$

$$h_\theta(x_i) = \theta_0 + \theta_1 x_1 + \theta_2 x_2^2 + \theta_3 x_3^3$$

L1 Regularization

$$mse = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - h_\theta(x_i)\right)^2 + \lambda \sum_{i=1}^{n}|\theta_i|$$

$$h_\theta(x_i) = \theta_0 + \theta_1 x_1 + \theta_2 x_2^2 + \theta_3 x_3^3$$

Ridge regression adds “*squared magnitude*” of coefficient as penalty term to the loss function

Lasso Regression (Least Absolute Shrinkage and Selection Operator) adds “*absolute value of magnitude*” of coefficient as penalty term to the loss function.

How to Prevent Overfitting or UnderFitting

1. Cross-validation. Cross-validation is a powerful preventative measure against **overfitting**. ...
2. Train with **more** data. It won't work every time, but training with **more** data can help algorithms detect the signal better. ...
3. Remove features. ...
4. Early stopping. ...
5. Regularization. ...
6. Ensembling.