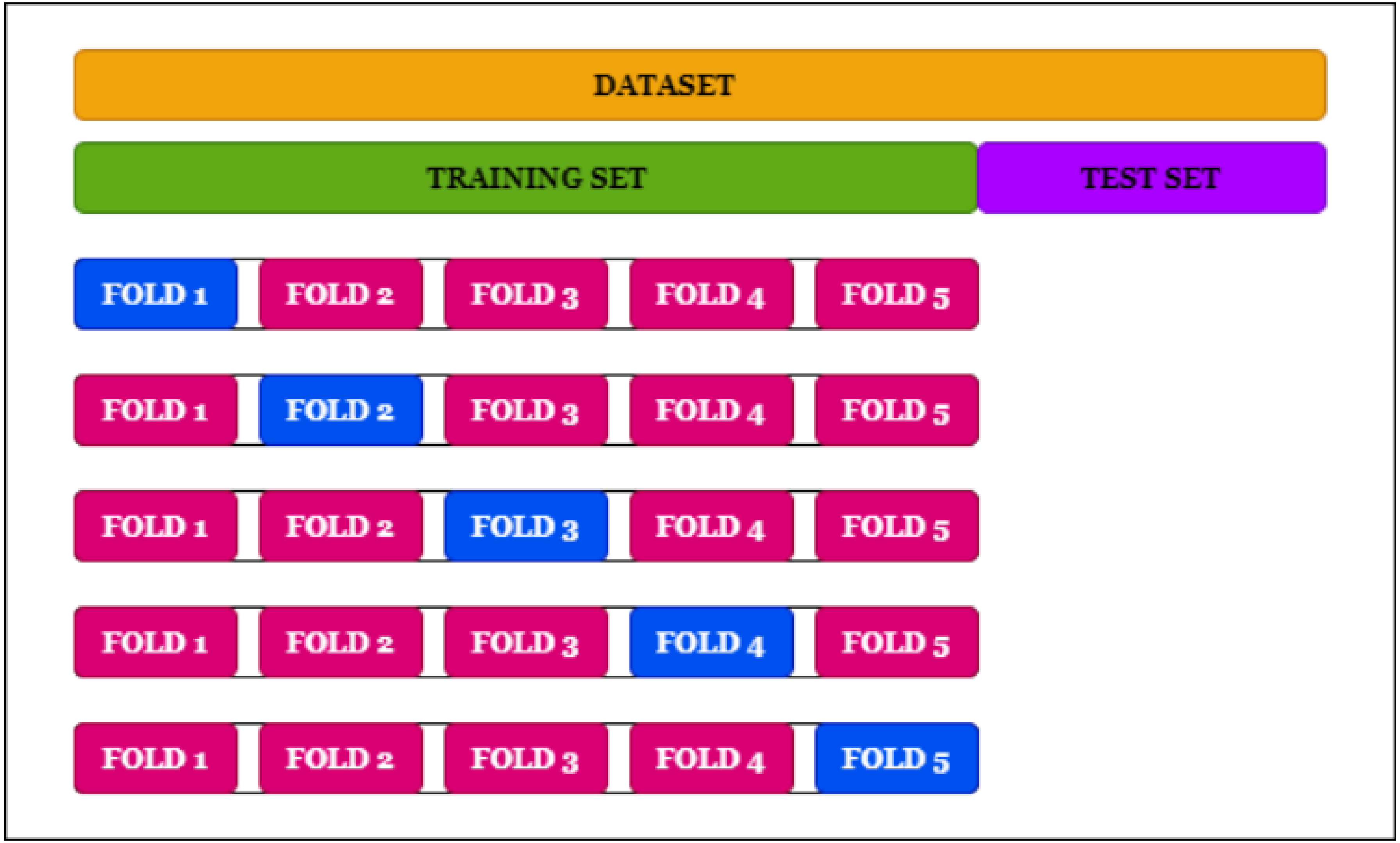
# Cross validation, Bias and Variance

# K fold Cross-validation



DATASET

TRAINING SET TEST SET

FOLD 1 FOLD 2 FOLD 3 FOLD 4 FOLD 5

FOLD 1 FOLD 2 FOLD 3 FOLD 4 FOLD 5

FOLD 1 FOLD 2 FOLD 3 FOLD 4 FOLD 5

FOLD 1 FOLD 2 FOLD 3 FOLD 4 FOLD 5

FOLD 1 FOLD 2 FOLD 3 FOLD 4 FOLD 5

# k-Fold Cross-Validation

- Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.
- The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into.
- As such, the procedure is often called k-fold cross-validation.
- When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation.
- Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data.

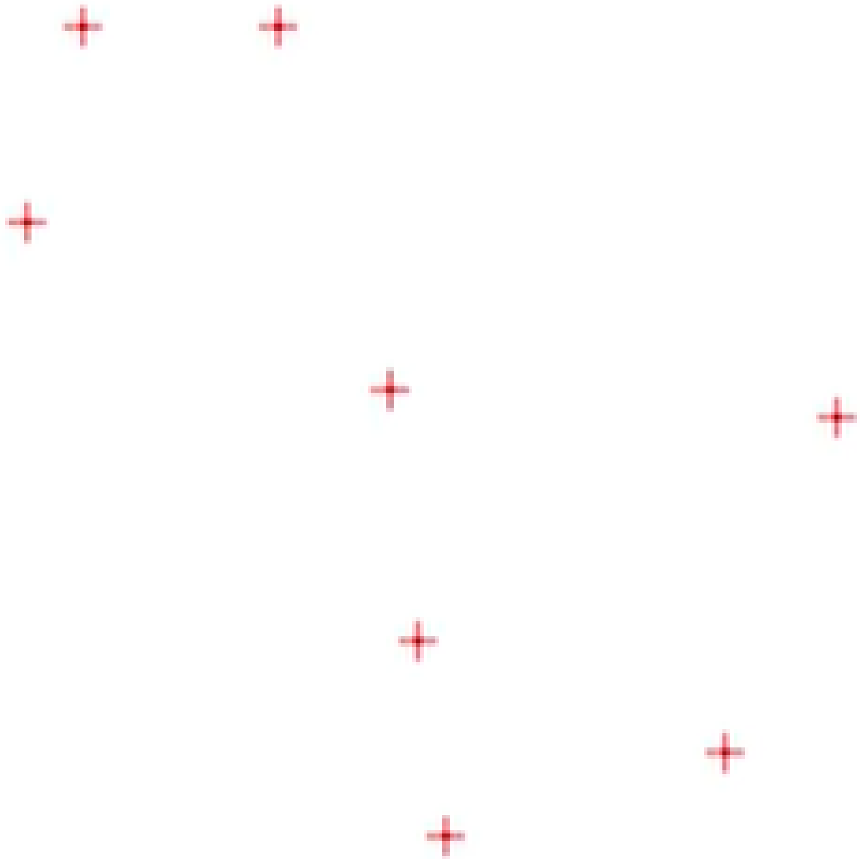
# k-Fold Cross-Validation

- It is a popular method because it is simple to understand and because it generally results in a less biased model.
- The general procedure is as follows:
  - 1. Shuffle the dataset randomly.
  - 2. Split the dataset into k groups
  - 3. For each unique group:
    - a. Take the group as a test data set
    - b. Take the remaining groups as a training data set
    - c. Fit a model on the training set and evaluate it on the test set
    - d. Retain the evaluation score (eg. Accuracy) and discard the model
  - 4. Take the mean of (all k )evaluation scores which results in accuracy.

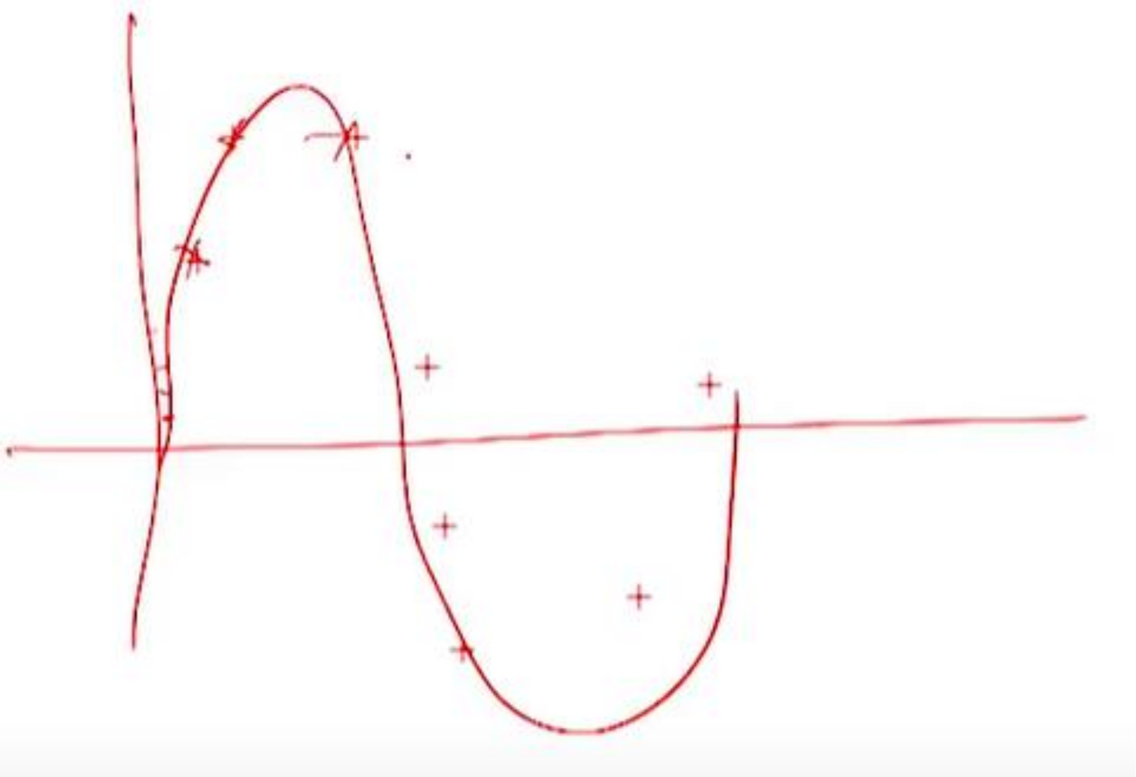
# k-Fold Cross-Validation

- This method is a good choice when we have a minimum amount of data and we get sufficiently big difference in quality or different optimal parameters between folds.
- As a general rule, we choose k=5 or k=10, as these values have been shown empirically to yield test error estimates that suffer neither from excessively high bias nor high variance

# Given set of points



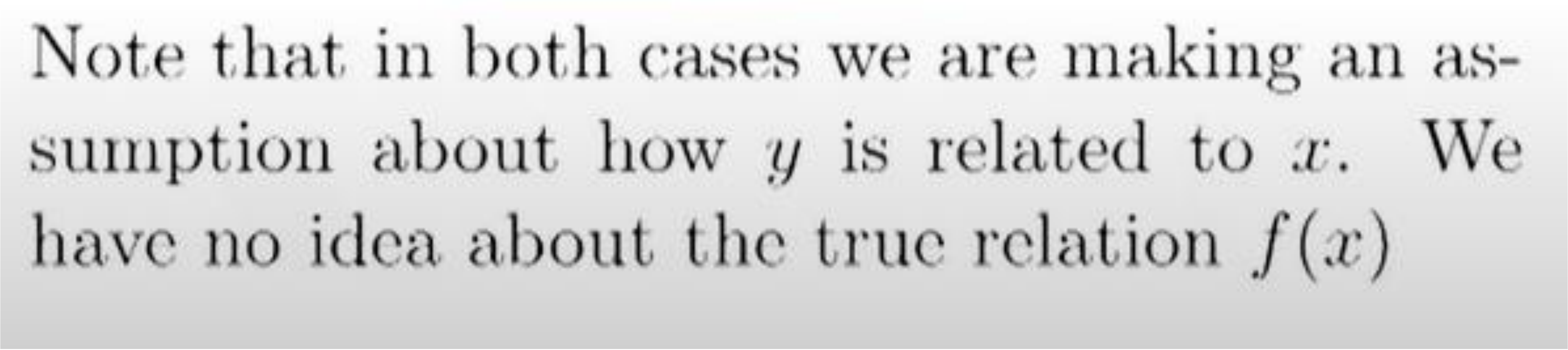
# This is actual function for this data ( which we never know for real world problem)



## Let us consider the problem of fitting a curve through a given set of points

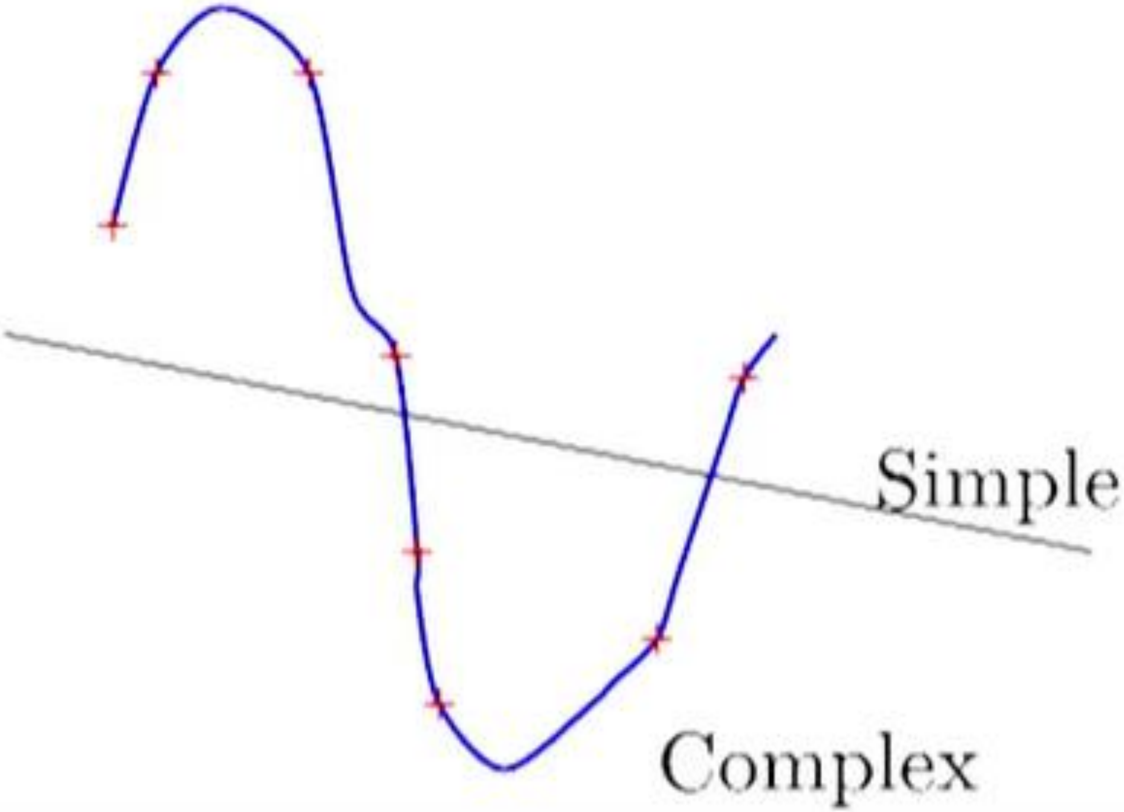
- We consider two models :

$$\begin{array}{cl} \begin{matrix} Simple \\ (degree{:}1) \end{matrix} & y = \hat{f}(x) = w_1 x + w_0 \\ \begin{matrix} Complex \\ (degree{:}25) \end{matrix} & y = \hat{f}(x) = \displaystyle\sum_{i=1}^{25} w_i x^i + w_0 \end{array}$$



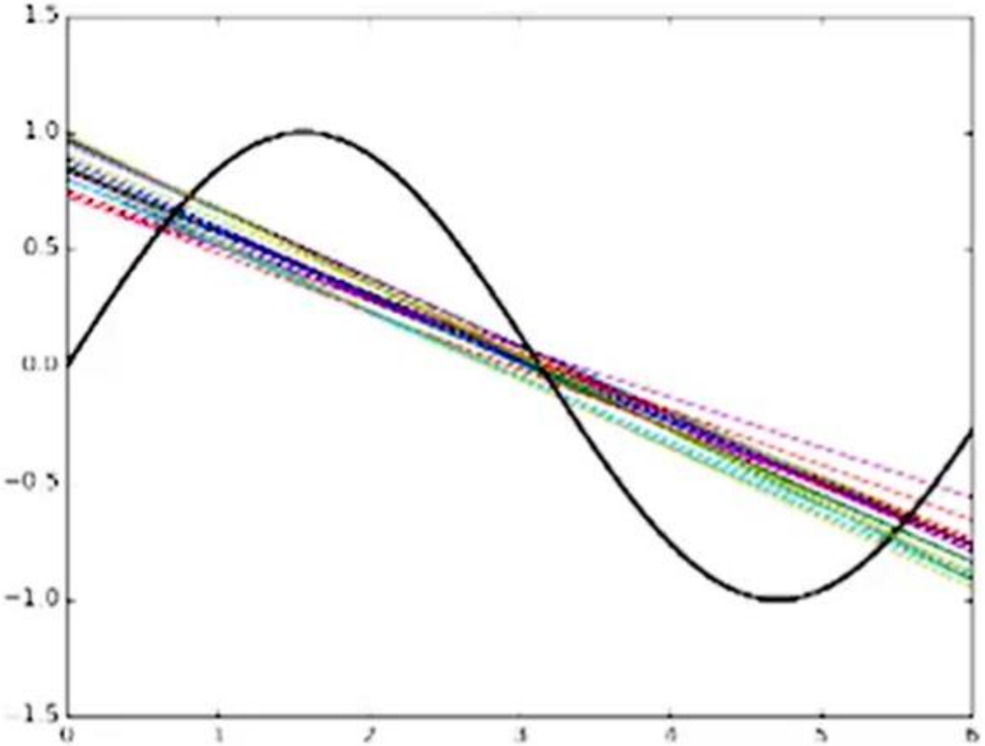
Note that in both cases we are making an assumption about how $y$ is related to $x$. We have no idea about the true relation $f(x)$

We sample 25 points from the training data and train a simple and a complex model

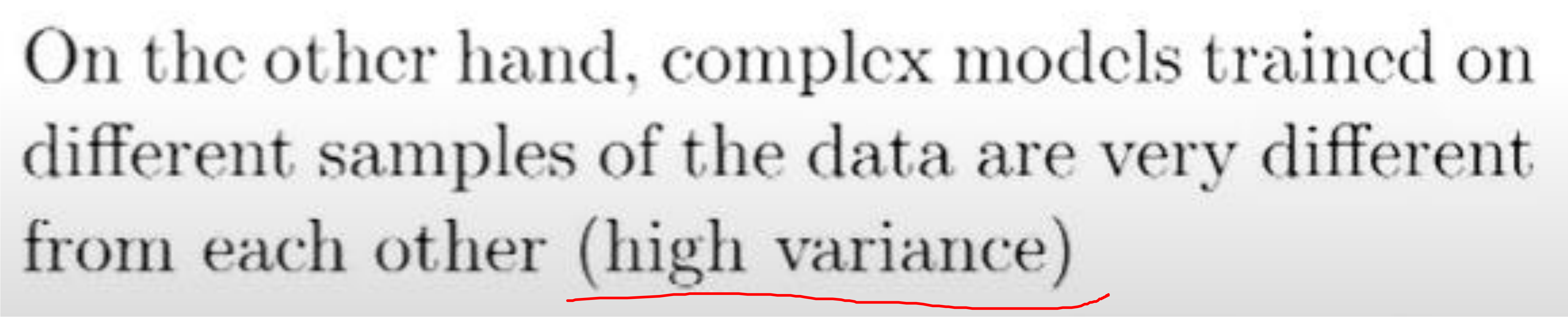


We repeat the process ‘$k$’ times to train multiple models (each model sees a different sample of the training data)

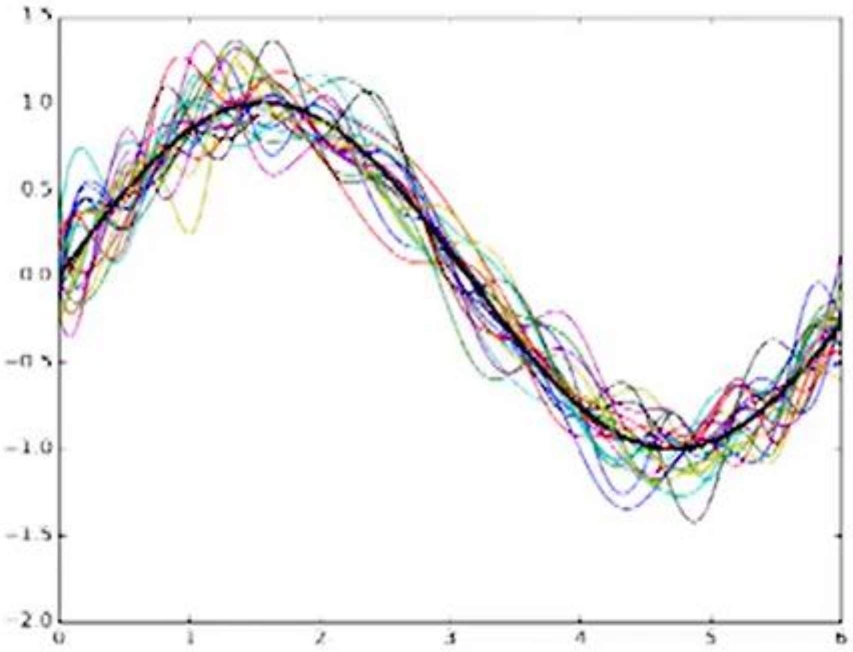
Simple models trained on different samples of the data do not differ much from each other (low variance)



However they are very far from the true sinusoidal curve (under fitting) (High Bias)

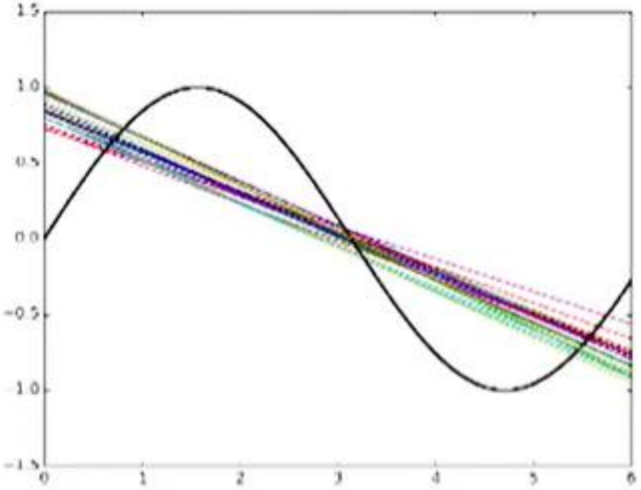


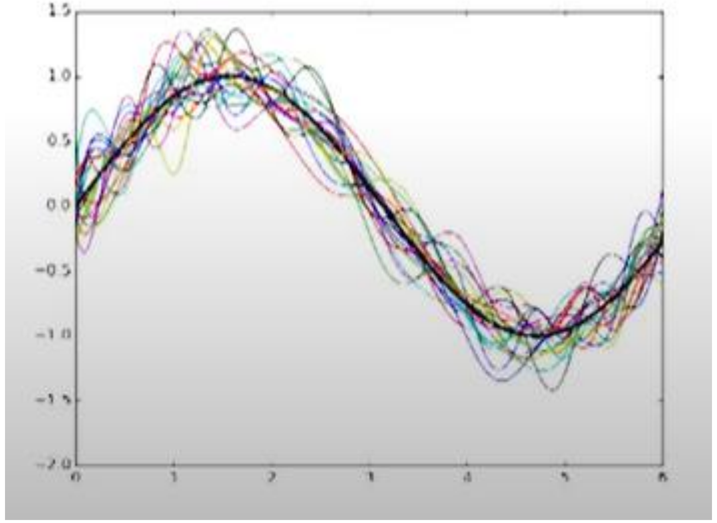
On the other hand, complex models trained on different samples of the data are very different from each other (high variance)



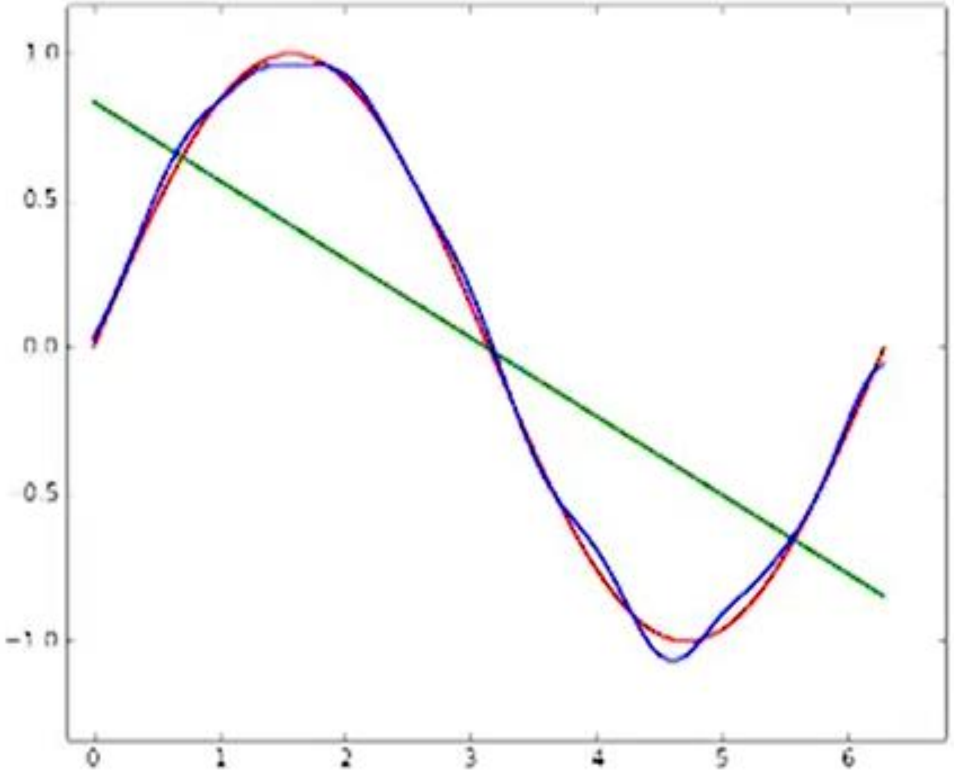
However they are very near to true sinusoidal curve (low Bias)

- Simple model: high bias, low variance
- Complex model: low bias, high variance





- There is always a trade-off between the bias and variance



Green Line: Average value of $\hat{f}(x)$ for the simple model

Blue Curve: Average value of $\hat{f}(x)$ for the complex model

Red Curve: True model $(f(\mathrm{x}))$

# Bias

$$\text{Bias}\ (\hat{f}(x)) = E[\hat{f}(x)] - f(x)$$

$E[\hat{f}(x)]$ is the average (or expected) value of the model

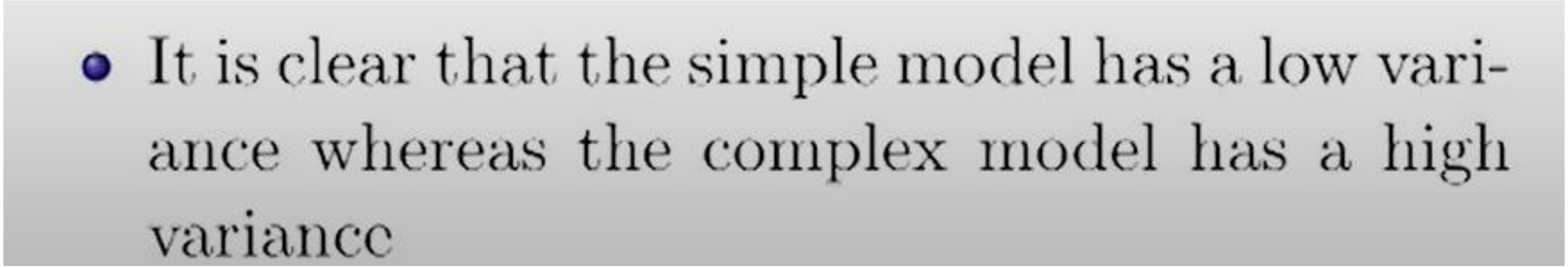
We can see that for the simple model the average value (green line) is very far from the true value $f(x)$ (sinusoidal function)

Mathematically, this means that the simple model has a high bias

On the other hand, the complex model has a low bias

# Variance

Tells how much the different prediction models (trained on different samples of the given data) differ from each other



- It is clear that the simple model has a low variance whereas the complex model has a high variance

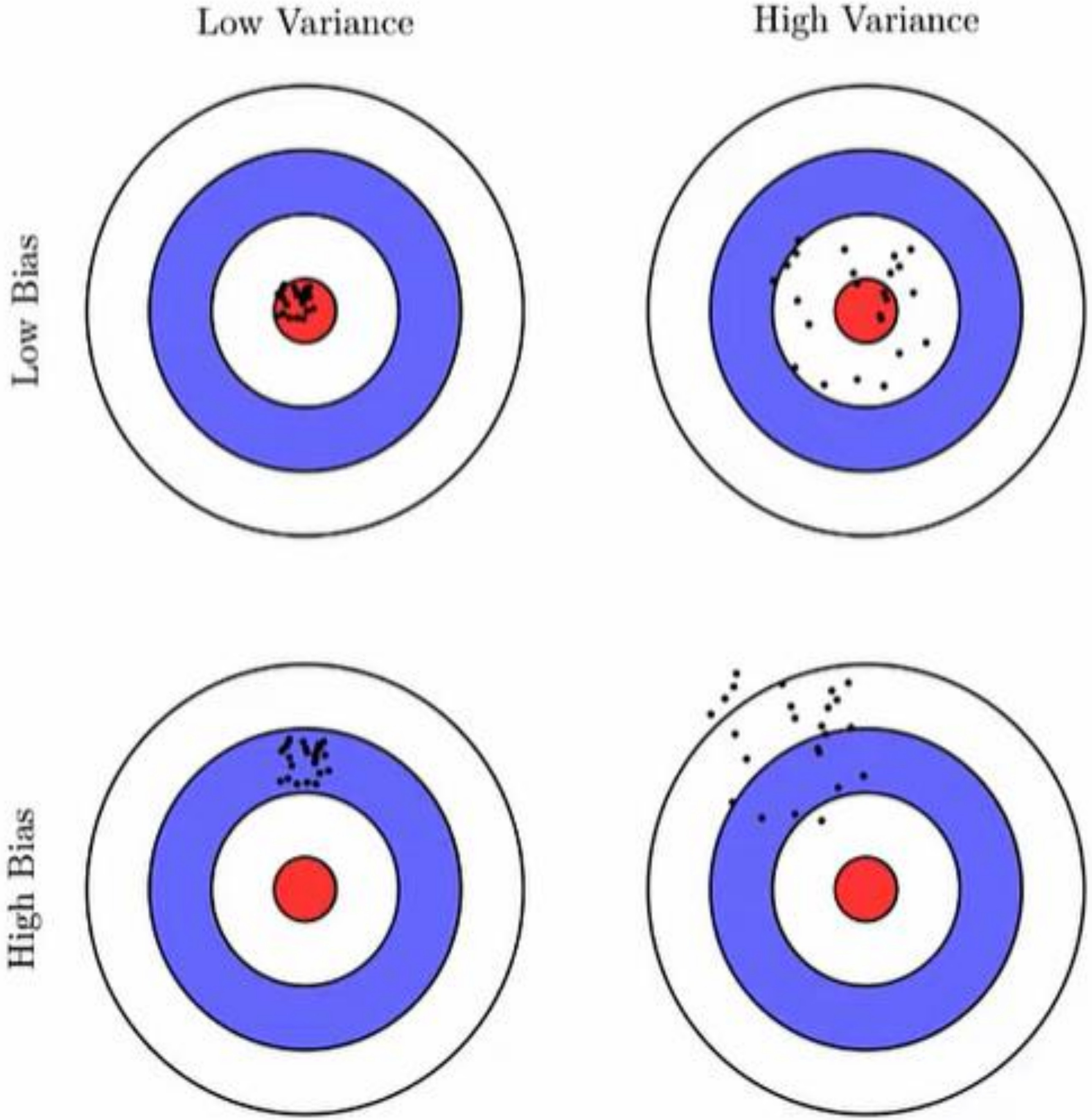
# Bias

- “Bias is error introduced in your model due to over simplification of machine learning algorithm.”
- It can lead to underfitting. When you train your model at that time model makes simplified assumptions to make the target function easier to understand.
- Low bias machine learning algorithms — Decision Trees, k-NN and SVM
- High bias machine learning algorithms — Linear Regression, Logistic Regression

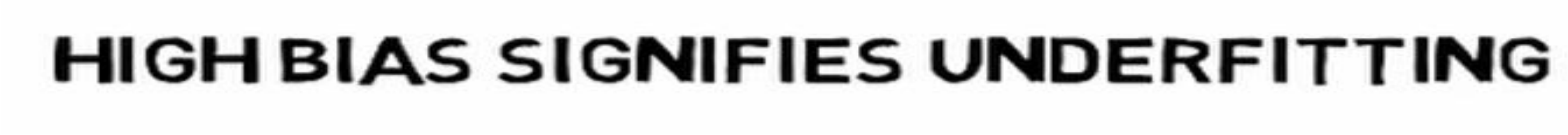
# Variance

- “Variance is error introduced in your model due to complex machine learning algorithm,
-  Your model learns noise also from the training data set and performs bad on test data set.”
- It can lead to overfitting.
- Normally, as you increase the complexity of your model, you will see a reduction in error due to lower bias in the model.
- However, this only happens till a particular point. As you continue to make your model more complex, you end up overfitting your model and hence your model will start suffering from high variance.

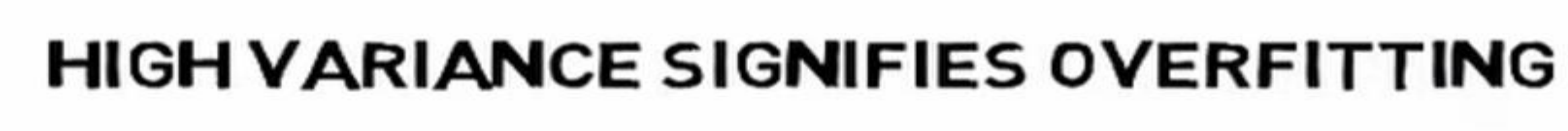
# BULL'S EYE DIAGRAM



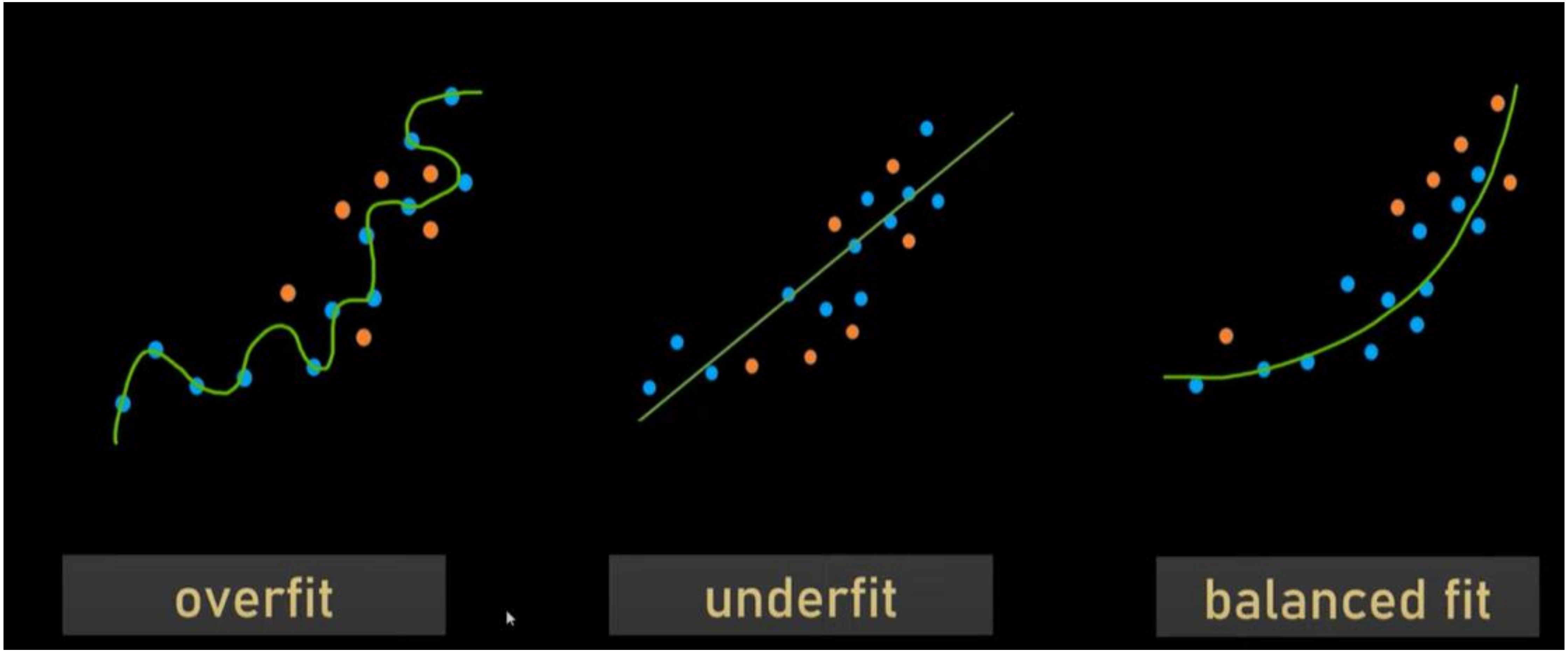
# Underfitting and Overfitting



HIGH BIAS SIGNIFIES UNDERFITTING



HIGH VARIANCE SIGNIFIES OVERFITTING



overfit

underfit

balanced fit

The goal of any supervised machine learning algorithm is to have low bias and low variance to achieve good prediction performance.

- Less features → underfitting
- Too many features → overfitting
- Only significant features → balanced fit (obtained by dimensionality reduction (PCA))