What is generalization

In supervised learning, the main goal is to use training data to build a model that will be able to make accurate predictions based on new, unseen data, which has the same characteristics as the initial training set. This is known as generalization.

Generalization error

To train a machine learning model, you split the dataset into 3 sets: training, validation, and testing. we train your models using the training data, then we compare and tune them using the evaluation results on the validation set, and in the end, evaluate the performance of your best model on the testing set. The error rate on new cases is called the generalization error (or out-of-sample error)

A model's generalization error (also known as a prediction error) can be expressed as the sum of three very different errors: Bias error, variance error, and irreducible error.

Note: The irreducible error occurs due to the noisiness of the data itself. The only way to reduce this part of the error is to clean up the data (e.g., fix the data sources, such as broken sensors, or detect and remove outliers).

Bias error

Characteristics of a high-bias model include:

- Failure to capture proper data trends
- Potential towards underfitting
- More generalized/overly simplified
- High error rate

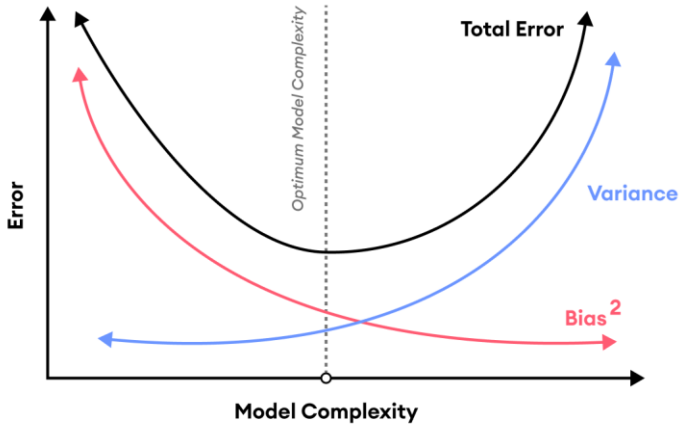
Variance error

A high-variance model typically has the following qualities:

- Noise in the data set
- Potential towards overfitting
- Complex models

Bias-variance tradeoff

> Bias/variance in machine learning relates to the problem of simultaneously minimizing two error sources (bias error and variance error).



In order to find a balance between underfitting and overfitting (the best model possible), you need to find a model which minimizes the total error.

Total Error = Bias²+ Variance + Irreducible Error

What is underfitting

Underfitting occurs when a model is not able to make accurate predictions based on training data and hence, doesn't have the capacity to generalize well on new data.

Machine learning models with underfitting tend to have poor performance both in training and testing sets (like the child who learned only addition and was not able to solve problems related to other basic arithmetic operations both from his math problem book and during the math exam).

Underfitting models usually have high bias and low variance.

What is overfitting

A model is considered overfitting when it does extremely well on training data but fails to perform on the same level on the validation data (like the child who memorized every math problem in the problem book and would struggle when facing problems from anywhere else).

Models that are overfitting usually have low bias and high variance.

How to detect underfitting

1) **Training and test loss:** If the model is underfitting, the loss for both training and validation will be considerably high. In other words, for an underfitting dataset, the training and the validation error will be high.

2) **Oversimplistic prediction graph:** If a graph with the data points and the fitted curve is plotted, and the classifier curve is oversimplistic, then, most probably, your model is underfitting. In those cases, a more complex model should be tried out.

How to avoid underfitting

There are several things you can do to prevent underfitting in AI and machine learning models:

1) **Train a more complex model** – Lack of model complexity in terms of data characteristics is the main reason behind underfitting models. For example, you may have data with upwards of 100000 rows and more than 30 parameters. If you train data with the Random Forest model and set max depth (max depth determines the maximum depth of the tree) to a small number (for example, 2), your model will definitely be underfitting. Training a more complex model (in this respect, a model with a higher value of max depth) will help us solve the problem of underfitting.

2) **More time for training** - Early training termination may cause underfitting. As a machine learning engineer, you can increase the number of epochs or increase the duration of training to get better results.

3) **Eliminate noise from data** – Another cause of underfitting is the existence of outliers and incorrect values in the dataset. Data cleaning techniques can help deal with this problem.

4) **Adjust regularization parameters** - the regularization coefficient can cause both overfitting and underfitting models.

How to detect overfitting

Some of the techniques you can use to detect overfitting are as follows:

1) Use a resampling technique to estimate model accuracy. The most popular resampling technique is k-fold cross-validation. It allows you to train and test your model k-times on different subsets of training data and build up an estimate of the performance of a machine learning model on unseen data. The drawback here is that it is time-consuming and cannot be applied to complex models, such as deep neural networks.

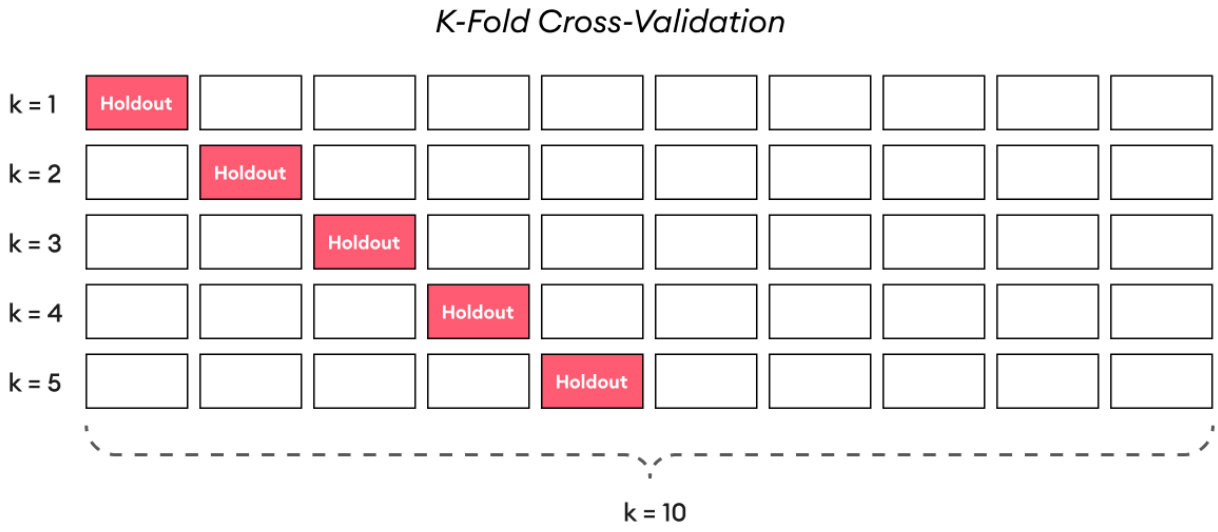


Figure 6. K-fold cross-validation: Image source

2) Hold back a validation set. Once a model is trained on the training set, you can evaluate it on the validation dataset, then compare the accuracy of the model in the training dataset and the validation dataset. A significant variance in these two results allows assuming that you have an overfitted model.

3) Another way to detect overfitting is by starting with a simplistic model that will serve as a benchmark. With this approach, if you try more complex algorithms, you will have a general understanding of whether the additional complexity for the model is worthwhile, if at all. This principle is known as *Occam's razor* test. This principle suggests that with all else being equal, simpler solutions to problems are preferred over more complex ones (if your model is not getting significantly better after using a much more complex model, it is preferable to use a simpler model).

How to prevent overfitting

Some of those methods are listed below.

1) **Adding more data** – Most of the time, adding more data can help machine learning models detect the "true" pattern of the model, generalize better, and prevent overfitting. However, this is not always the case, as adding more data that is inaccurate or has many missing values can lead to even worse results.

2) **Early stopping** – In iterative algorithms, it is possible to measure how the model iteration performance. Up until a certain number of iterations, new iterations improve the model. After that point, however, the model's ability to generalize can deteriorate as it begins to overfit the training data. Early stopping refers to stopping the training process before the learner passes that point.

3) **Data augmentation** – In machine learning, data augmentation techniques increase the amount of data by slightly changing previously existing data and adding new data points or by producing synthetic data from a previously existing dataset.

4) **Remove features** – You can remove irrelevant aspects from data to improve the model. Many characteristics in a dataset may not contribute much to prediction. Removing non-essential characteristics can enhance accuracy and decrease overfitting.

5) **Regularization** – Regularization refers to a variety of techniques to push your model to be simpler. The approach you choose will be determined by the model you are training. For example, you can add a penalty parameter for a regression (L1 and L2 regularization), prune a decision tree or use dropout on a neural network.

6) **Ensembling** – Ensembling methods merge predictions from numerous different models. These methods not only deal with overfitting but also assist in solving complex machine learning problems (like combining pictures taken from different angles into the overall view of the surroundings). The most popular ensembling methods are boosting and bagging.

- **Boosting** – In boosting method, you train a large number of weak learners (constrained models) in sequence, and each sequence learns from the mistakes of the previous sequence. Then you combine all weak learners into a single strong learner.

- **Bagging** is another technique to reduce overfitting. It trains a large number of strong learners (unconstrained models) and then combines them all in order to optimize their predictions.