Supervised Learning

As discussed, in this type of machine learning, the model is fed with some input and output (called labels), and it has find an optimal solution that maps the inputs and outputs correctly. Here, it “learns” how to map input and output by analysing and finding relations.

There are various steps to be followed when it comes to supervised learning:

- Determine the type of training examples:

It is important to know what kind of training data to be used so that we can build a relavent model.

- Gather the training set:

Now that we know what type of training examples we are taking, it is important to know what all to include in it so that it is accurate for the real-world problems the model is going to face.

- Determine the accuracy of input features that will be represented in the learned functions.
- Determine the structure of the learned function and the learning algorithm.
- Complete the design and test it by evaluating the accuracy of the learned function.

There are two important terms to be understood when going through this kind of machine learning:
- Bias
- Variance

These are prediction errors and it is important to know the trade-off between the two in order to build an efficient model that is accurate and relavent to real-world problems.

Bias

Refers to how biased the model is, or how small it thinks the real-time problem data is. It is usually biassed towards what it has learnt and is unwilling to change itself no matter how often it will be trained in the future.

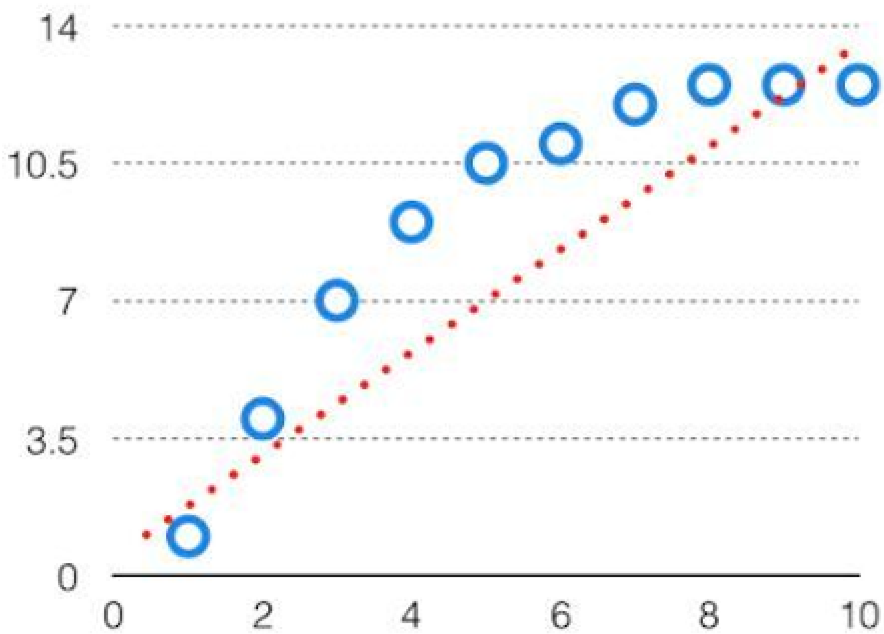
Given a dataset, the bias curve only meets a few data points on the data curve. It is generally a linear graph and it represents **underfitting**, which means the model has been trained over lesser number of situations, or not all real-world scenarios have been trained.

In simple words, a model with high bias:

- Has a *simple function*
- *Assumes*
- *Underfits* data

The graph representing this kind of prediction error is:

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$



Variance

Refers to how much the model varies when testing with different datasets, or how much the model tries to fit the training data provided. In this case, the model fits in every data point of a training data by drawing a curve that passes through all the points.

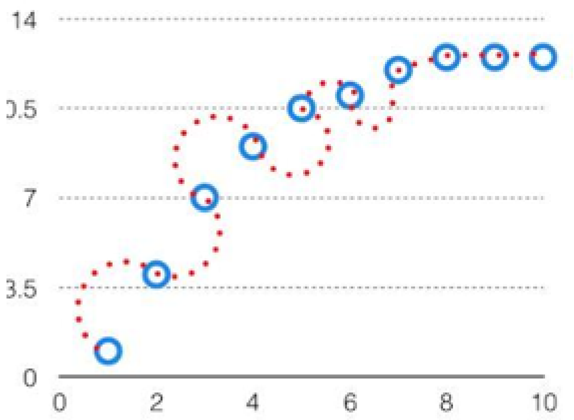
It may sound right, and why it could be an error. It is because, it depends a lot on the training data and does not know anything beyond it. In other words, the model is to rigid and sensitive to the training data. This is called **overfitting**.

In simple words, a model with high variance:

- Has a *complex function*
- *Sensitive*
- *Overfits* data

The graph given by this kind of prediction error is:

$$h_\theta(x) = g(\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4)$$



Issues of Supervised Learning

Supervised learning comes with some major drawbacks, which are listed as below:

1. Bias - Variance trade-off
2. Training data and function complexity
3. Dimensionality of input vectors
4. Degree of noise in output predicted

Bias - Variance Trade-off

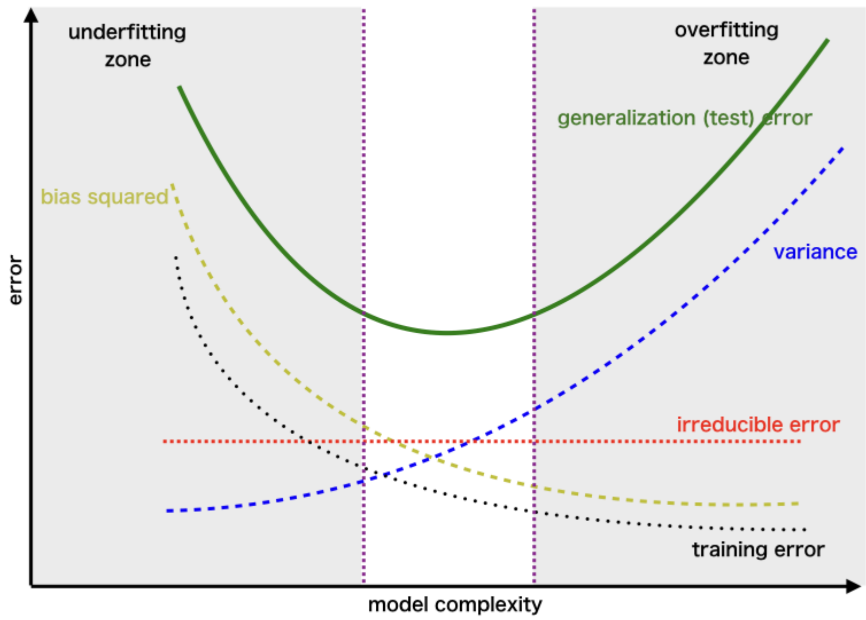
As of what we had already seen, we can infer that bias and variance are opposite to each other. Any attempt made to reduce the error due to one of these will increase the error due to the other. That is, if we attempt to reduce the bias of a model, it has to achieve higher variance, and vice versa.

The problem comes with how much to reduce such errors to achieve the maximum possible accuracy. The best possible way is to reduce the error (bias or variance) such that it gets reduced and the other doesn't grow. Basically creating a balance between the two. But it is highly challenging considering various parameters such as dataset size and change in accuracy.

There is a formula for determining the error of a model, and it goes like this:

$$\text{error} = \text{bias}^2 + \text{variance} + \text{irreducable error}$$

Here, irreducable error refers to those errors which can never be avoided or exists by nature. Such errors can not be removed by any error removing methods. Hence this term is constant, and the error of the model is dependent only on the bias and variance, which can be manipulated.



Training Data and Function Complexity

Did you know that the training data fed a supervised model is dependent on the function complexity of the model? First of all, what is function complexity?

Function Complexity: It defines how complex the function of a training model is in terms of the number of parameters it contains. The more the number of parameters, the more complex the function is.

Now, the training data is dependent on the complexity of the function because the amount of data a model uses or needs depends on its bias or variance. A model which has high bias only goes through a few data points and hence, a small dataset would be enough. Whereas, a model with high variance goes through every data point, and requires a large dataset.

Dimensionality of Input Vectors

Dimentionality of input vectors simply means the number of features present in a dataset. For a dataset of n features, data points are plotted on a nD plot.

Why could this be a problem? Lets say our data has many features but less samples. It means we are providing only a part of the possible inputs, and the model learns less. Also, it tries to go through every single datapoint, which becomes challenging as the number of dimensions increase. Hence, in such cases it is required feed a huge dataset, although the model learns with high variance.

Hence, it is required to get rid of features that harm the performance of the model in order to achieve maximum accuracy.

Degree of Noise in Output Predicted

The ultimate goal of any supervised learning model is to predict the relation between a given set of inputs and outputs. But there can be noises in the outputs provided, which eventually confuses the model and hence, the model makes bad predictions.

A noise is an unexpected error in data that is either caused naturally or by us, who provides the data. It could be due to recording false data, bad instruments or sensors which records data etc.

In this case, the model tries to find a meaningful patter between the input and output, which do not exist due to noise. Hence, the model ends up going through every data point to find a meaningful pattern, and *overfits*.