Dimensionality Reduction

Dimensionality Reduction

- The number of input features, variables, or columns present in a given dataset is known as dimensionality, and the process to reduce these features is called dimensionality reduction.

- A real-life dataset may contain a huge number of input features in various cases, which makes the predictive modeling task more complicated.

- Because it is very difficult to visualize or make predictions for the training dataset with a high number of features, for such cases, dimensionality reduction techniques are required to use.

Dimensionality reduction technique can be defined as, "It is a way of converting the higher dimensions dataset into lesser dimensions dataset ensuring that it provides similar information.“

These techniques are widely used in machine learning for obtaining a better fit predictive model while solving the classification and regression problems.

It is commonly used in the fields that deal with high-dimensional data, such as speech recognition, signal processing, bioinformatics, etc.

Dimensionality reduction helps to find out the most significant features and skips the rest.

It makes the data easy for plotting in 2D and 3D plots which leads to better human interpretation.

It can also be used for data visualization, noise reduction, cluster analysis, etc.

The Curse of Dimensionality

- Handling the high-dimensional data is very difficult in practice, commonly known as the *curse of dimensionality*.

- If the dimensionality of the input dataset increases, any machine learning algorithm and model becomes more complex. As the number of features increases, and the chance of overfitting also increases.

- If the machine learning model is trained on high-dimensional data, it becomes overfitted and results in poor performance.

- Hence, it is often required to reduce the number of features, which can be done with dimensionality reduction.

What is bias?

Bias is the difference between the average prediction of our model and the correct value which we are trying to predict.

Model with high bias pays very little attention to the training data and oversimplifies the model.

Variance

In machine learning, the term "variance" refers to the amount by which a model's predictions would change if it were trained on a different dataset. It is a measure of how sensitive the model is to the variations in the training data.

Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn’t seen before. As a result, such models perform very well on training data but has high error rates on test data

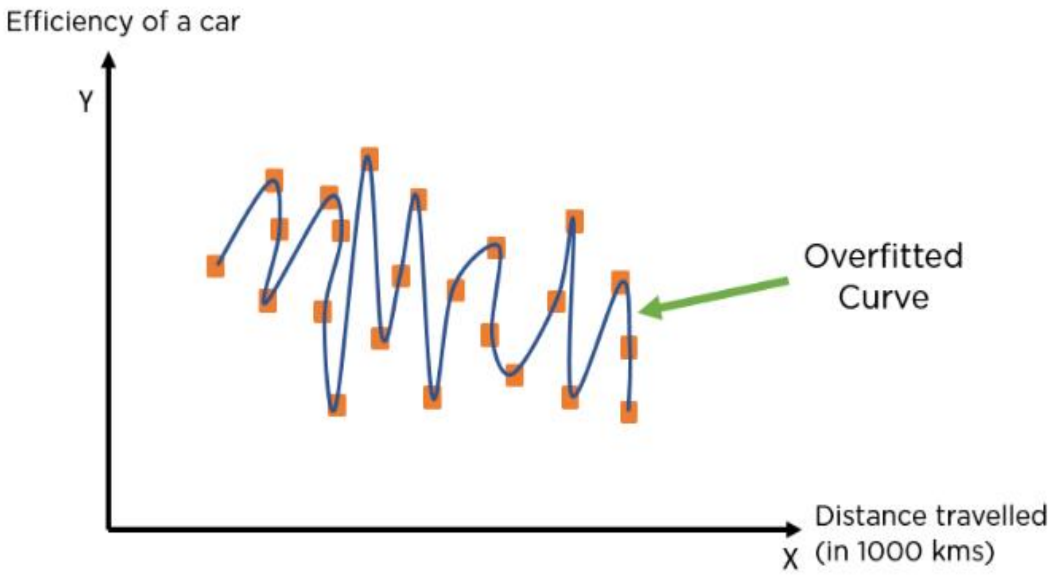
Overfitting in ML

When a model performs very well for training data but has poor performance with test data (new data), it is known as overfitting.

In this case, the machine learning model learns the details and noise in the training data such that it negatively affects the performance of the model on test data.

Overfitting is the main problem that occurs in supervised learning.

The overfitted model has **low bias** and **high variance.**



The concept of the overfitting can be understood by the above graph of the linear regression output:

As we can see from the above graph, the model tries to cover all the data points present in the scatter plot. It may look efficient, but in reality, it is not so.

Because the goal of the regression model to find the best fit line, but here we have not got any best fit, so, it will generate the prediction errors.

How to avoid the Overfitting in Model

Both overfitting and underfitting cause the degraded performance of the machine learning model. But the main cause is overfitting, so there are some ways by which we can reduce the occurrence of overfitting in our model.

- **Cross-Validation**
- **Training with more data**
- **Removing features**
- **Ensembling**

Underfitting

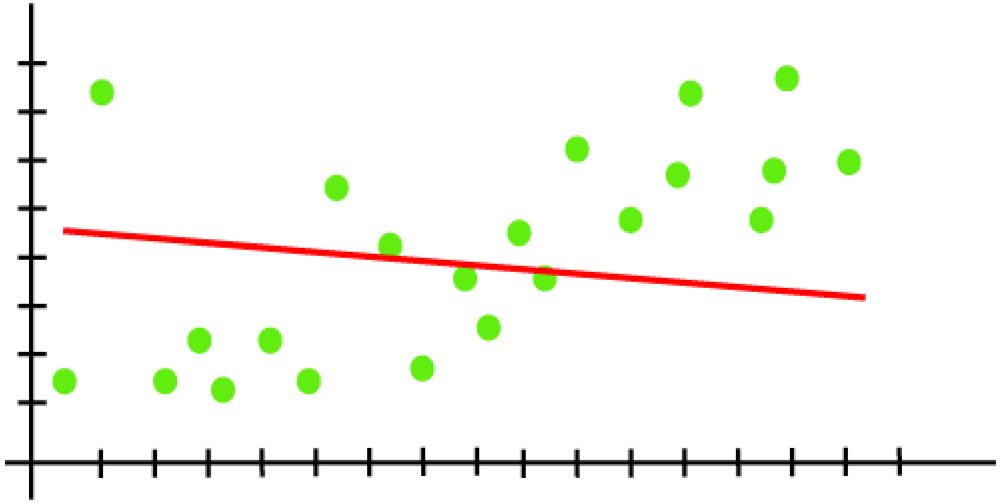
In supervised learning, **underfitting** happens when a model unable to capture the underlying pattern of the data. These models usually have high bias and low variance.

It happens when we have very less amount of data to build an accurate model or when we try to build a linear model with a nonlinear data.

More formally, your hypothesis about data distribution is wrong and too simple — for example, your data is quadratic and your model is linear.

This situation is also called **high bias**. This means that your algorithm can do accurate predictions, but the initial assumption about the data is incorrect.

We can understand the underfitting using below output of the linear regression model:

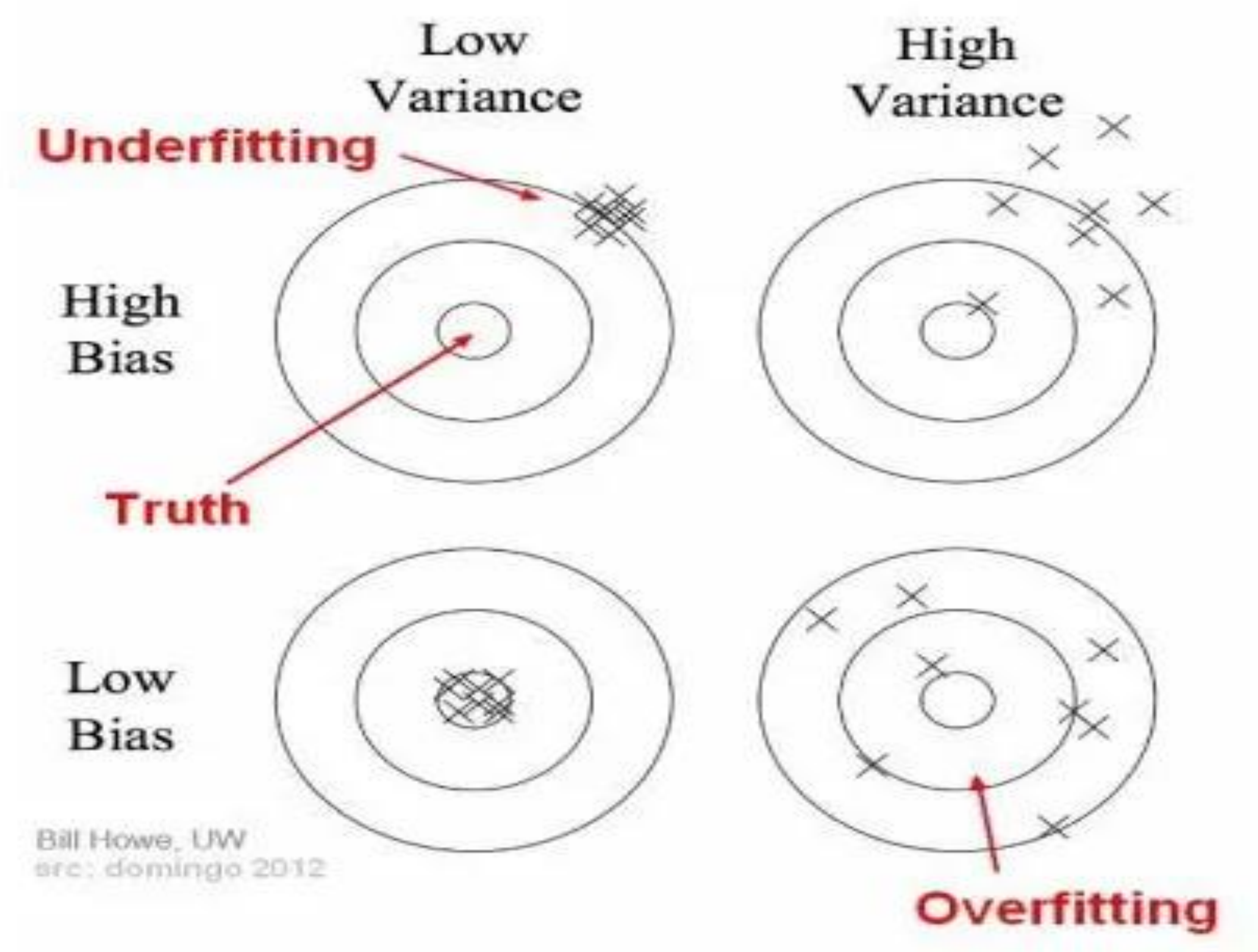


Reasons for Underfitting

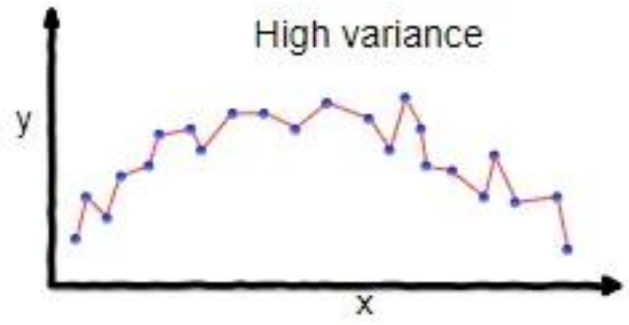
- Data used for training is not cleaned and contains noise (garbage values) in it
- The model has a high bias
- The size of the training dataset used is not enough
- The model is too simple

Ways to Tackle Underfitting

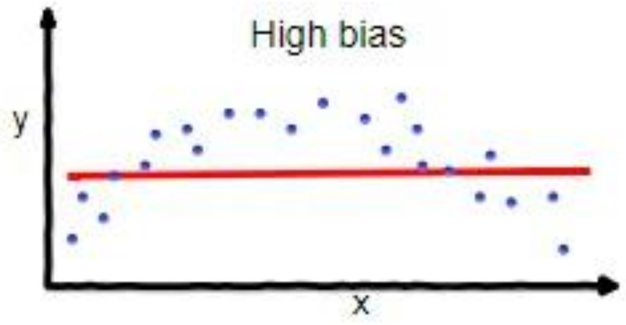
- Increase the number of features in the dataset
- Increase model complexity
- Reduce noise in the data
- Increase the duration of training the data



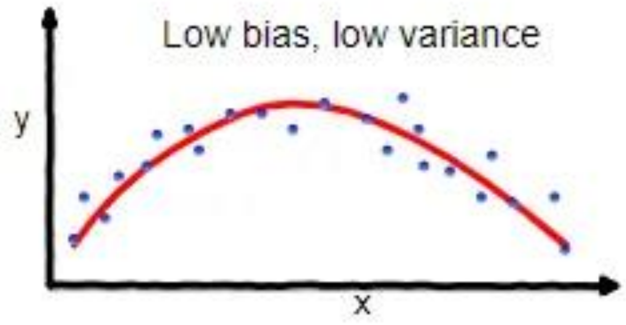
In the above diagram, center of the target is a model that perfectly predicts correct values. As we move away from the bulls-eye our predictions become get worse and worse.



overfitting



underfitting



Good balance

In supervised learning, **underfitting** happens when a model unable to capture the underlying pattern of the data.

In supervised learning, **overfitting** happens when our model captures the noise along with the underlying pattern in data. It happens when we train our model a lot over noisy dataset.

Bias Variance Tradeoff

- low bias, low variance — is a good result, just right.

- low bias, **high variance** — **overfitting** — the algorithm outputs very different predictions for similar data.

- **high bias**, low variance — **underfitting** — the algorithm outputs similar predictions for similar data, but predictions are wrong (algorithm “miss”).

- high bias, high variance — very bad algorithm. You will most likely never see this.

Benefits of applying Dimensionality Reduction

Some benefits of applying dimensionality reduction technique to the given dataset are given below:

- By reducing the dimensions of the features, the space required to store the dataset also gets reduced.

- Less Computation training time is required for reduced dimensions of features.

- Reduced dimensions of features of the dataset help in visualizing the data quickly.

- **I**t removes the redundant features (if present)

Disadvantages of dimensionality Reduction

There are also some disadvantages of applying the dimensionality reduction, which are given below:

- Information loss: Dimensionality reduction methods aim to retain the most important information while discarding less relevant or redundant features. However, in the process of reducing dimensions, some information is inevitably lost.

- Overfitting: Dimensionality reduction techniques can inadvertently introduce overfitting, especially if the reduction is performed without considering the target variable. By compressing the data, the reduced feature space may not fully capture the underlying patterns and relationships between the features and the target variable.