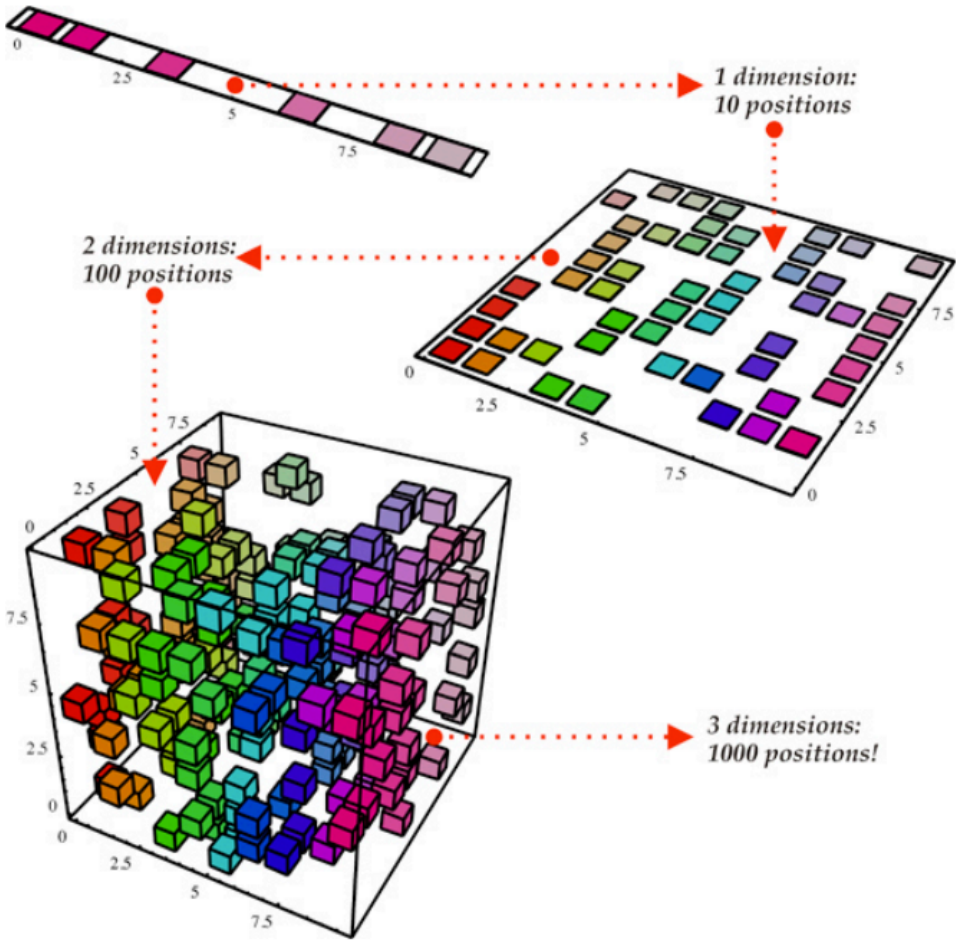
Dimensionality Reduction for Machine Learning

- Have you ever worked on a dataset with more than a Hundred features? How about over 50,000 features?
- We have, and let us tell you it's a very challenging task, especially if you don't know where to start! Having a high number of variables is both a boon and a curse. It's great that we have loads of data for analysis, but it is challenging due to size.
- Using dimensionality reduction techniques, of course. You can use this concept to reduce the number of features in your dataset without having to lose much information and keep (or improve) the model's performance.

What is Dimensionality Reduction?

- Dimensionality reduction is the process of reducing the number of random variables under consideration, by obtaining a set of principal variables.



Why is Dimensionality Reduction required?

- Higher Dimensionality causes Curse of Dimensionality. so to reduce Curse of Dimensionality we can use Dimensionality Reduction.
- Fewer dimensions lead to less computation/training time.
- Some algorithms do not perform well when we have large dimensions. So reducing these dimensions needs to happen for the algorithm to be useful.

- It takes care of multicollinearity by removing redundant features. For example, you have two variables – ‘time spent on a treadmill in minutes’ and ‘calories burnt’. These variables are highly correlated as the more time you spend running on a treadmill, the more calories you will burn. Hence, there is no point in storing both as just one of them does what you require.
- It helps in visualizing data. As discussed earlier, it is very difficult to visualize data in higher dimensions so reducing our space to 2D or 3D may allow us to plot and observe patterns more clearly.
- If we have more features than observations than we run the risk of massively overfitting our model — this would generally result in terrible out of sample performance.

Curse of Dimensionality

- The curse of dimensionality basically means that the error increases with the increase in the number of features.
- It refers to the fact that algorithms are harder to design in high dimensions and often have a running time exponential in the dimensions

Dimensionality Reduction Techniques