Under fitting and Over fitting

Data Science with Python

Introduction

- Machine Learning model, we actually talk about how well it performs and its accuracy which is known as prediction errors. Let us consider that we are designing a machine learning model.
- A model is said to be a good machine learning model if it generalizes any new input data from the problem domain in a proper way.
- This helps us to make predictions about the future data, that the data model has never seen.
- Now, suppose we want to check how well our machine learning model learns and generalizes to the new data.

generalizes

- The main goal of each machine learning model is **to generalize well**. Here **generalization** defines the ability of an ML model to provide a suitable output by adapting the given set of unknown input.
- It means after providing training on the dataset, it can produce reliable and accurate output.
- Hence, the underfitting and overfitting are the two terms that need to be checked for the performance of the model and whether the model is generalizing well or not.

Bias and Variance

- **Bias** and **variance** are two key sources of error in machine learning models that directly impact their performance and generalization ability.
- **Bias:** Assumptions made by a model to make a function easier to learn. It is actually the error rate of the training data. When the error rate has a high value, we call it High Bias and when the error rate has a low value, we call it low Bias.
- **Variance:** The error rate of the testing data is called variance. When the error rate has a high value, we call it High variance and when the error rate has a low value, we call it Low variance.
- **Noise:** Noise is unnecessary and irrelevant data that reduces the performance of the model.

Bias

- It is the error that happens when a machine learning model is too simple and doesn’t learn enough details from the data.
- It’s like assuming all birds can only be small and fly, so the model fails to recognize big birds like ostriches or penguins that can’t fly and get biased with predictions.
- These assumptions make the model easier to train but may prevent it from capturing the underlying complexities of the data.
- High bias typically leads to **underfitting**, where the model performs poorly on both training and testing data because it fails to learn enough from the data.
- **Example**: A linear regression model applied to a dataset with a non-linear relationship.

Variance:

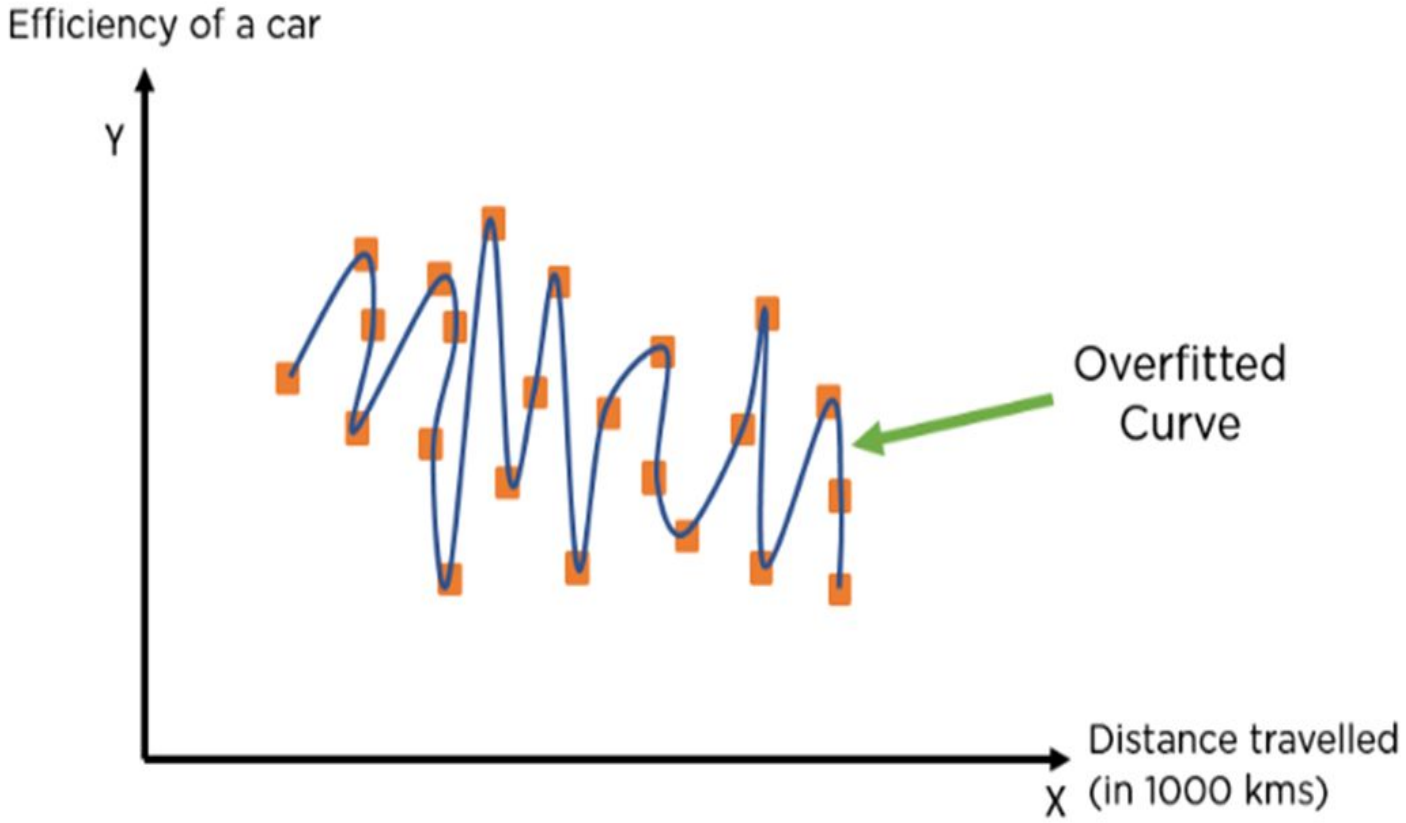
- Error that happens when a machine learning model learns too much from the data, including random noise.
- A high-variance model learns not only the patterns but also the noise in the training data, which leads to poor generalization on unseen data.
- High variance typically leads to **overfitting**, where the model performs well on training data but poorly on testing data.

What is Overfitting?

- 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.
- i.e., model tries to cover all the data points or more than the required data points present in the given dataset. Because of this, the model starts caching noise and inaccurate values present in the dataset, and all these factors reduce the efficiency and accuracy of the model.
- Overfitting can happen due to low bias and high variance.

Over fitting

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.



Reasons for Overfitting

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

Ways to Tackle Overfitting

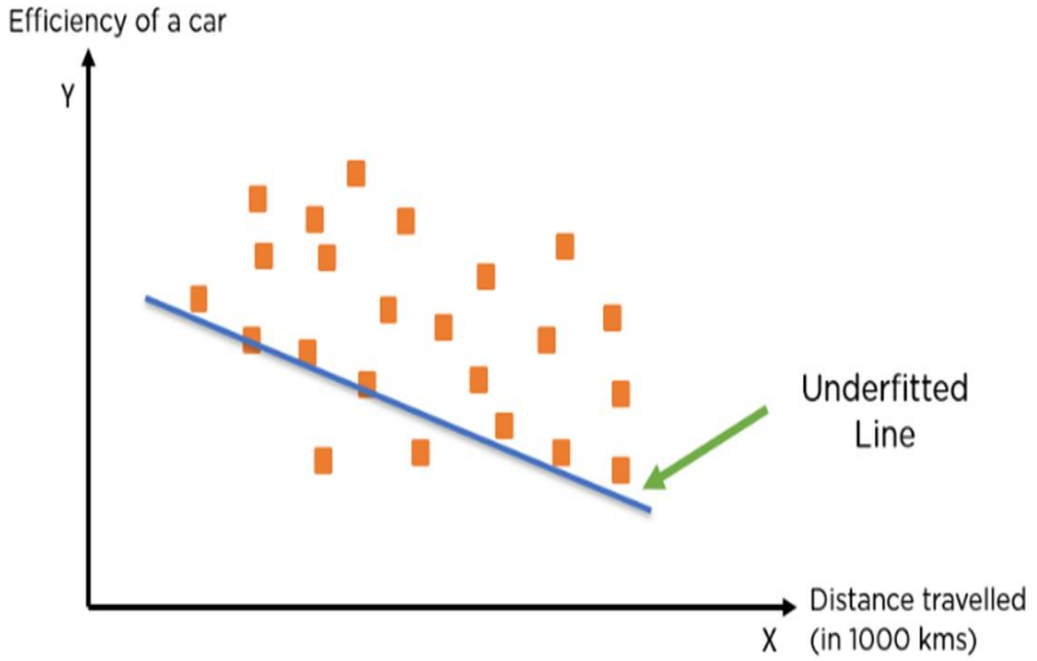
- Using K-fold cross-validation
- Using Regularization techniques such as Lasso and Ridge
- Training model with sufficient data
- Adopting ensembling techniques
- **Early stopping the training**
- **Training with more data**
- **Removing features**

What is Underfitting?

- When a model has not learned the patterns in the training data well and is unable to generalize well on the new data, it is known as underfitting. An underfit model has poor performance on the training data and will result in unreliable predictions. Underfitting occurs due to high bias and low variance.
- In the case of underfitting, the model is not able to learn enough from the training data, and hence it reduces the accuracy and produces unreliable predictions.

Under fitting

As we can see from the above diagram, the model is unable to capture the data points present in the plot.



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

How to avoid under fitting:

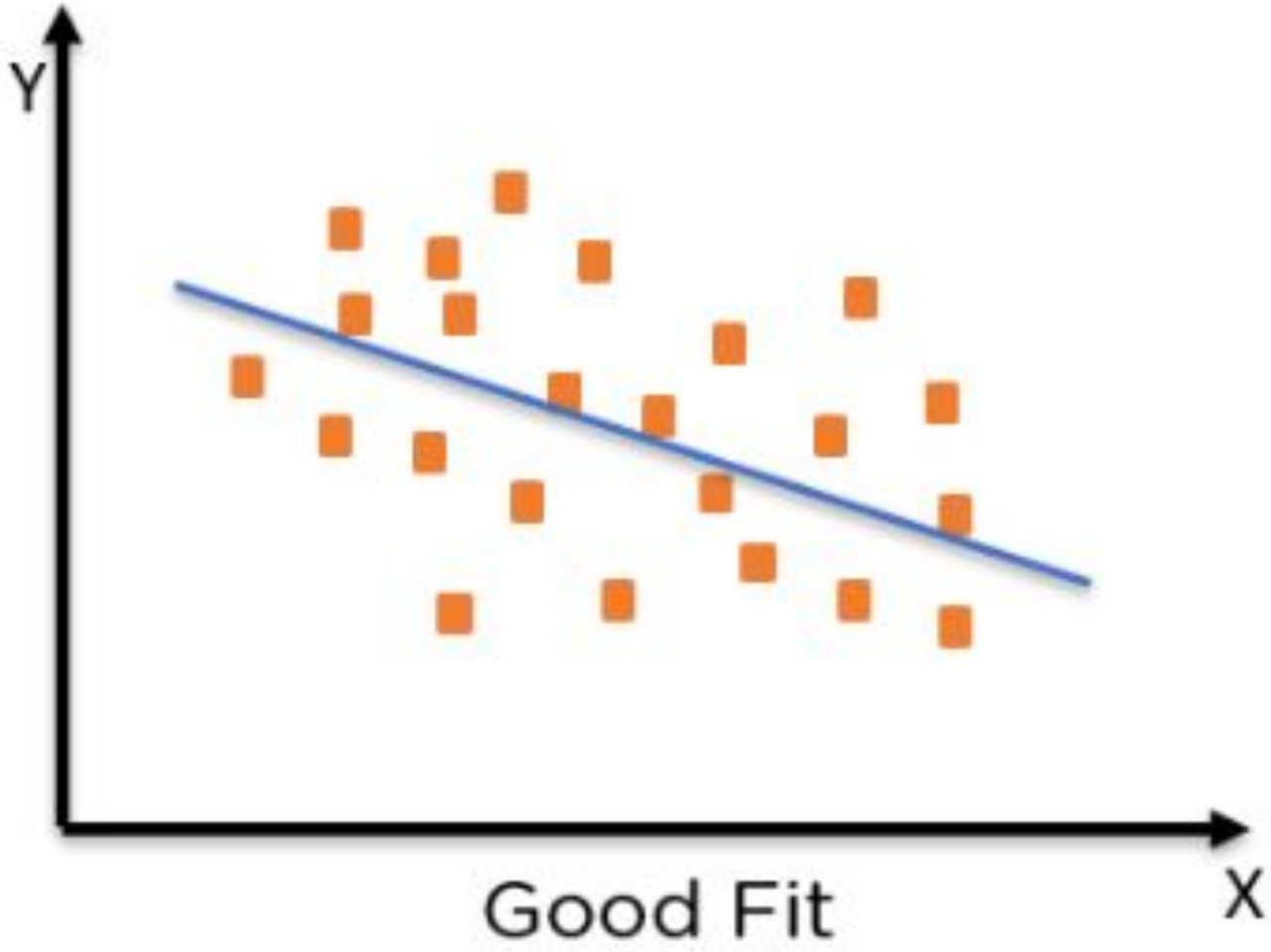
- By increasing the training time of the model.
- By increasing the number of features.
- Increase model complexity
- Reduce noise in the data
- Increase the number of epochs or increase the duration of training the data to get better results.

What Is a Good Fit

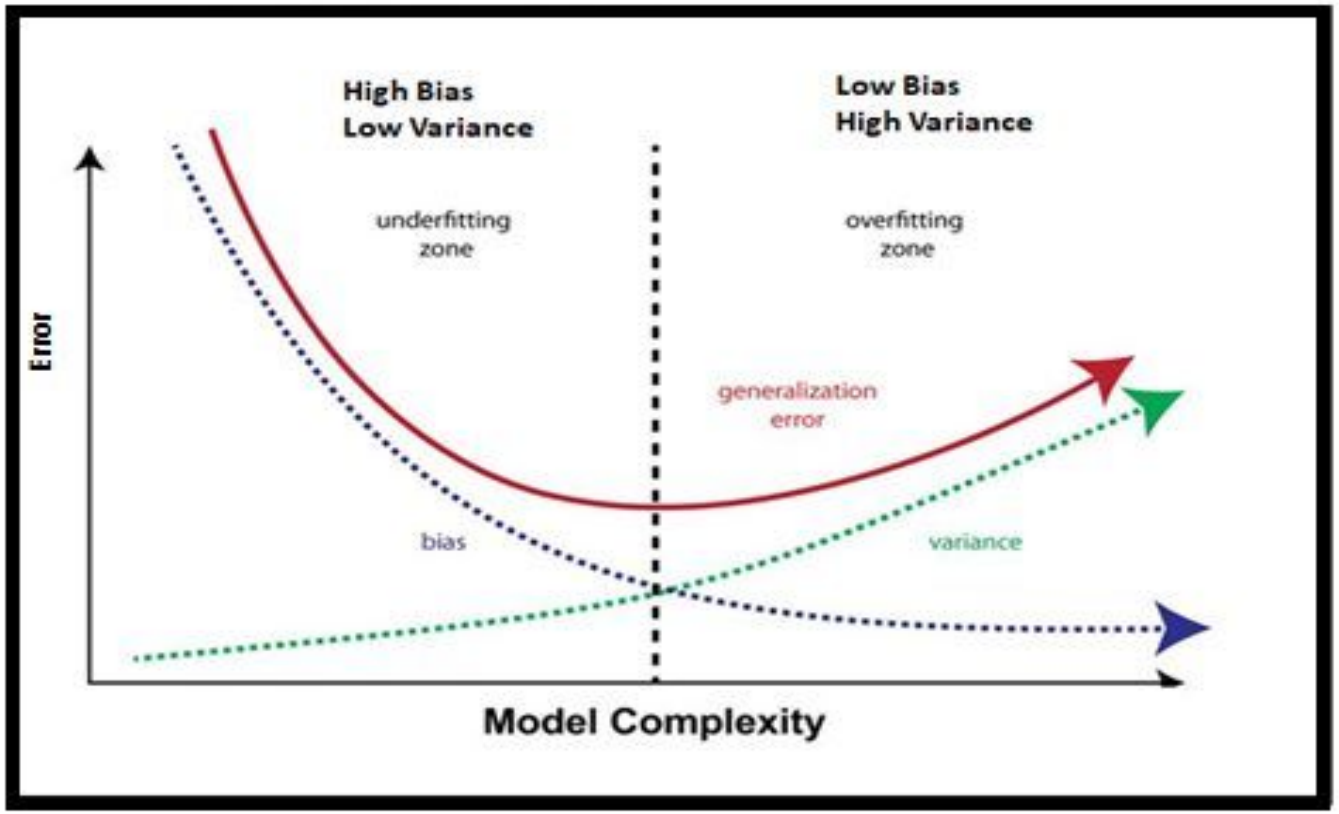
- To find the good fit model, you need to look at the performance of a machine learning model over time with the training data.
- As the algorithm learns over time, the error for the model on the training data reduces, as well as the error on the test dataset.
- If you train the model for too long, the model may learn the unnecessary details and the noise in the training set and hence lead to overfitting.
- In order to achieve a good fit, you need to stop training at a point where the error starts to increase.

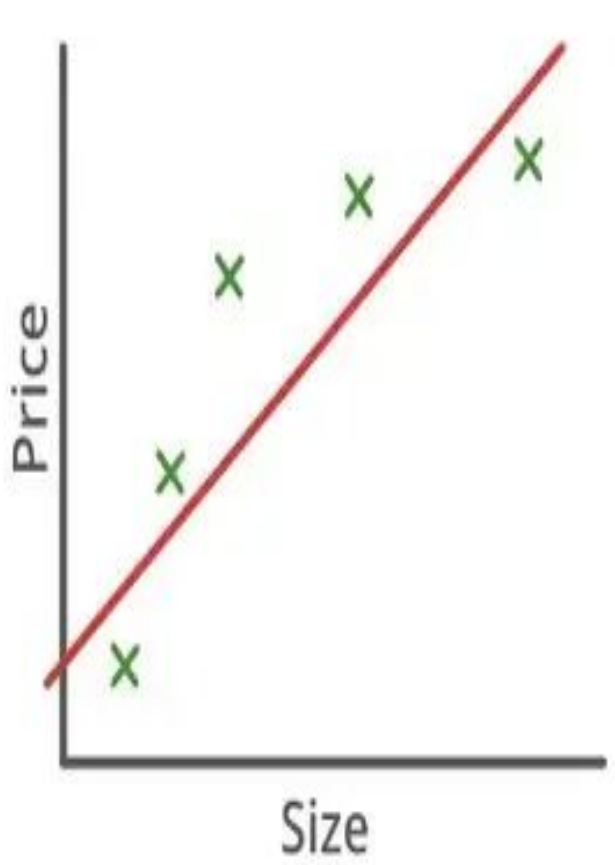
Best Fit

An ideal model strikes a balance with low bias and low variance, capturing the overall pattern without overreacting to noise. For instance, a smooth second-degree polynomial fits the data well without being overly complex.



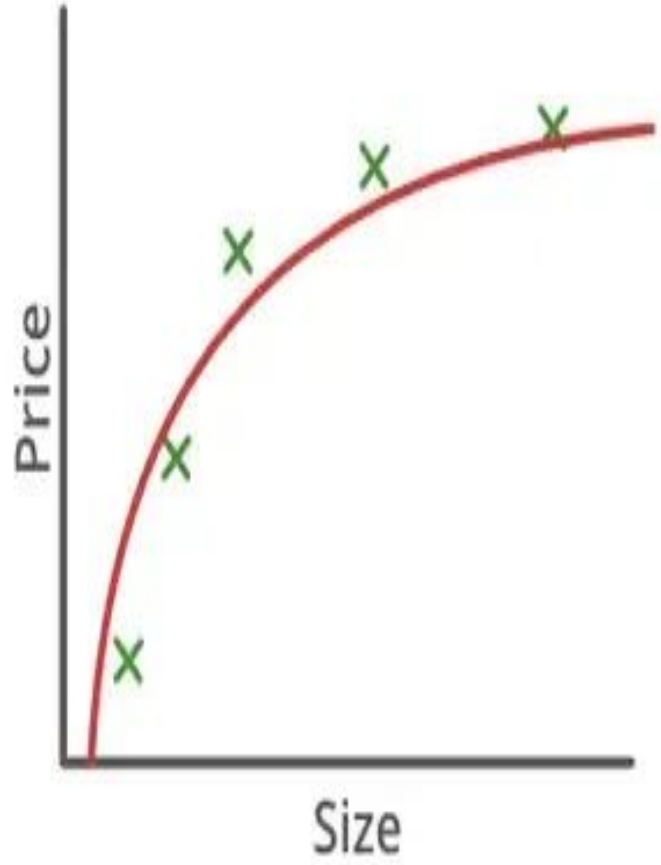
- we have to keep it in our mind is “Increasing the bias will decrease the variance. Increasing the variance will decrease bias,
  So **Bias** and **Variance** are indirectly proportional to each other.
- You can observe from the above visualization, that the model complexity has increased after a certain point, your model becomes overfitting.





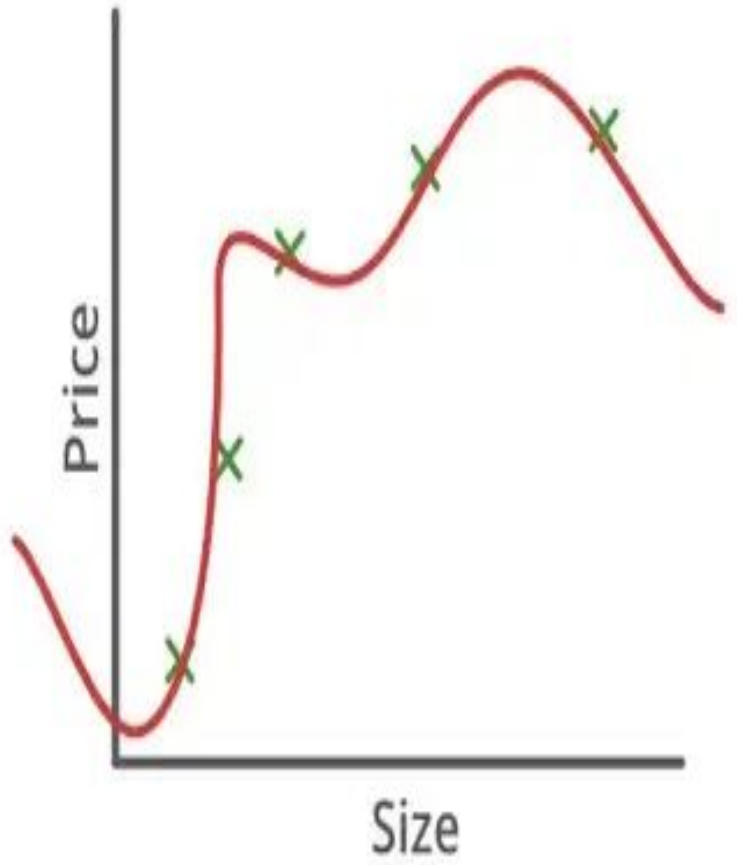
$\theta_0 + \theta_1 x$

High Bias
(Underfitting)



$\theta_0 + \theta_1 x + \theta_2 x^2$

Low Bias, Low Variance
(Goodfitting)



$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

High Variance
(Overfitting)

Types of Model Selection

- There are 2 major techniques in model selection, as mentioned earlier this is a **mathematical model and** patterns are extracted from the given dataset.
- Resampling
- Probabilistic

Resampling

- These are simple techniques just rearranging data samples and inspecting that the model performs good or bad with the data set.

Random Split

- Data has been randomly sampled.
- % percentage of data has been used for Training ,Testing and Validation sets.
- It usually prevent a biased sampling of data.
- Test set used test for model evaluation.
- Validation set used for in model selection, Not used for any other process

Time-Based

- Data has been split based on time-wise.
- Here model is trained till a particular date/time.
- Testing on the future dates/time in iterative form.
- Training window keeps increasing shifting by one day and Test set also reduces by a day.
- This method is stabilizing the model and prevents the model from overfitting.

K-Fold Cross-Validation

- Here we're randomly shuffling the dataset and then splitting it into k groups.
- On iterating over each group, by one group to be considered as a test set and all other groups are clubbed together considered as the training set.
- At the end of the process, one has k has different results on k different test groups in folder.
- The highest score will be best model.

Bootstrap

- Most powerful ways to obtain a stabilized model.
- The first step is to select a sample size, followed by a sample data point selected from the original dataset.
- And this data point is added to the bootstrap sample.
- After the addition, the sample needs to be put back into the original sample.
- Repeating this process for N times, where N is the sample size.