# ML Assignment
# Lab 07

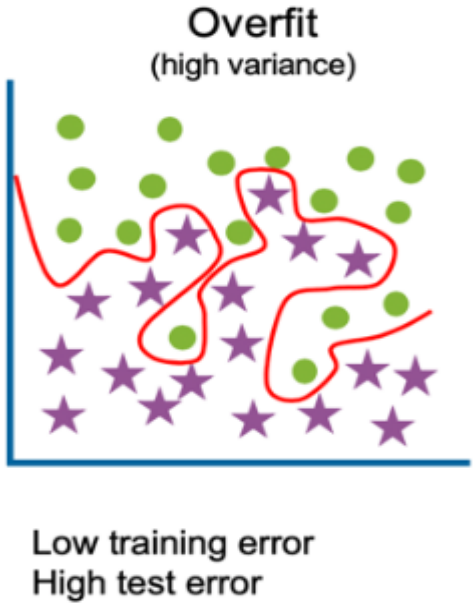
---

21BDA09 Dhinsha T

**4. What is overfitting? How to overcome overfitting in an ML model?**

Ref: https://medium.com/slalom-build/what-is-overfitting-exactly-4f3a43f01c34

A statistical model is said to be overfitted when we feed it a lot more data than necessary.When a model fits more data than it actually needs, it starts catching the noisy data and inaccurate values in the data. As a result, the efficiency and accuracy of the model decrease.



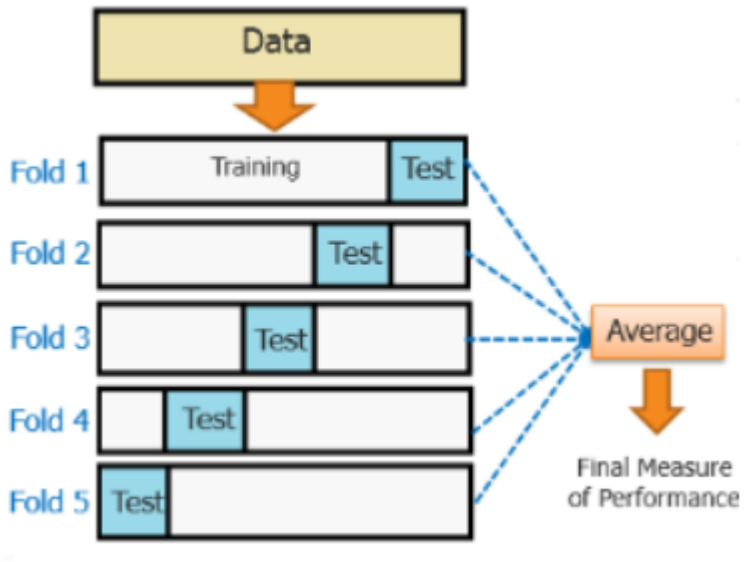

More formally, your hypothesis about data distribution is wrong and too complex for example, your data is linear and your model is a high-degree polynomial. This situation is also called high variance. This means that your algorithm can't do accurate predictions changing the input data only a little, the model output changes very much.

There are several techniques to avoid overfitting in Machine Learning

1. **Cross-Validation**

Cross-validation is a powerful preventative measure against overfitting.

The idea behind this is to use the initial training data to generate mini train-test-splits, and then use these splits to tune your model.In a standard k-fold validation, the data is partitioned into k-subsets also known as folds. After this, the algorithm is trained iteratively on k-1 folds while using the remaining folds as the test set, also known as holdout fold.




The cross-validation helps us to tune the hyperparameters with only the original training set. It basically keeps the test set separately as a true unseen data set for selecting the final model. Hence, avoiding overfitting altogether.

2. **Training With More Data**

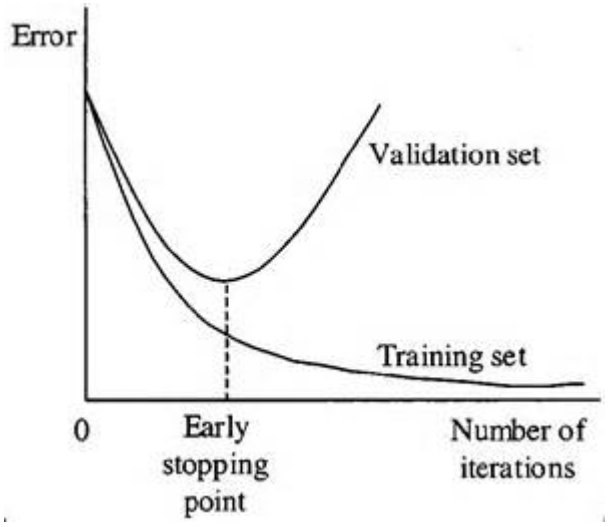
This technique might not work every time. Training with more data can help algorithms detect the signal better.  When we are training the model with more data, we have to make sure the data is clean and free from randomness and inconsistencies.

3. **Removing Features**

Although some algorithms have an automatic selection of features. For a significant number of those who do not have a built-in feature selection, we can manually remove a few irrelevant features from the input features to improve the generalization.
One way to do it is by deriving a conclusion as to how a feature fits into the model. It is quite similar to debugging the code line-by-line.
In case if a feature is unable to explain the relevancy in the model, we can simply identify those features. We can even use a few feature selection heuristics for a good starting point.

4. **Early Stopping**

Early stopping refers to stopping the training process before the learner passes that point.
When you're training a learning algorithm iteratively, you can measure how well each iteration of the model performs.
Up until a certain number of iterations, new iterations improve the model. After that point, however, the model's ability to generalize can weaken as it begins to overfit the training data.
Today, this technique is mostly used in deep learning while other techniques (e.g. regularization) are preferred for classical machine learning.




5. **Regularization**

Regularization refers to a broad range of techniques for artificially forcing your model to be simpler.Oftentimes, the regularization method is a hyperparameter as well, which means it can be tuned through cross-validation.

6. **Ensembling**

Ensembles are machine learning methods for combining predictions from multiple separate models. There are a few different methods for ensembling, but the two most common are:

Bagging attempts to reduce the chance of overfitting complex models.

- It trains a large number of "strong" learners in parallel.
- A strong learner is a model that's relatively unconstrained.
- Bagging then combines all the strong learners together in order to "smooth out" their predictions.

Boosting attempts to improve the predictive flexibility of simple models.

- It trains a large number of "weak" learners in sequence.
- A weak learner is a constrained model (i.e. you could limit the max depth of each decision tree).
- Each one in the sequence focuses on learning from the mistakes of the one before it.
- Boosting then combines all the weak learners into a single strong learner.

While bagging and boosting are both ensemble methods, they approach the problem from opposite directions.

Bagging uses complex base models and tries to "smooth out" their predictions, while boosting uses simple base models and tries to "boost" their aggregate complexity.