Machine Learning 2

Ramesh Kestur, IIITB

Rameshkestur@iiitb.ac.in

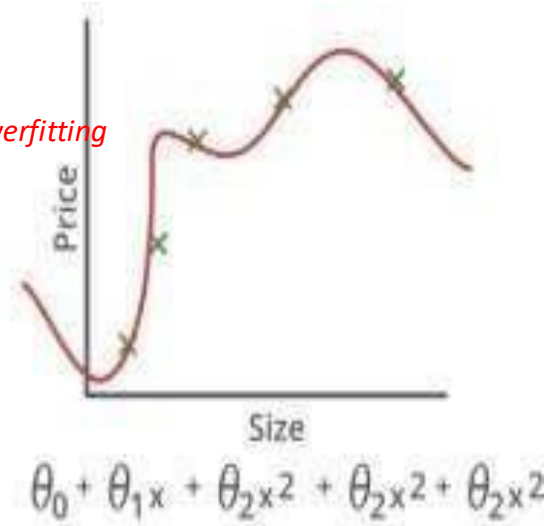
Public profile:
https://in.linkedin.com/in/rameshkestur
https://scholar.google.co.in/citations?user=P28Shk8AAAAJ&hl=en
https://www.iiitb.ac.in/faculty/ramesh-kestur

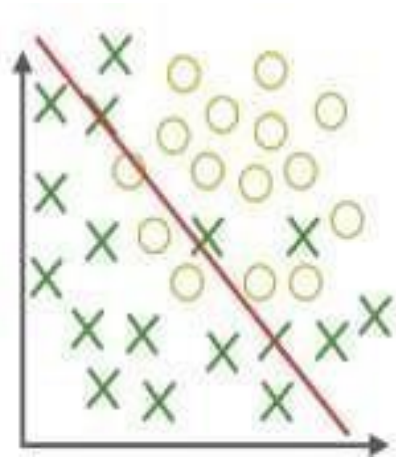
20-Dec-23

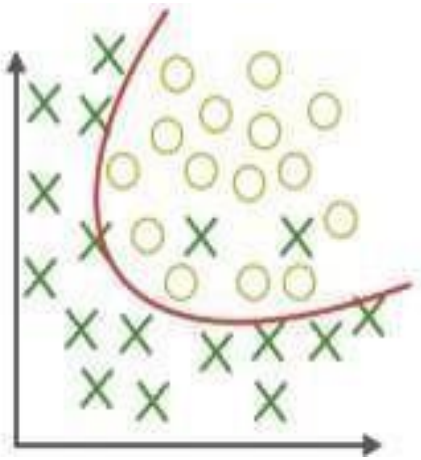
Topics

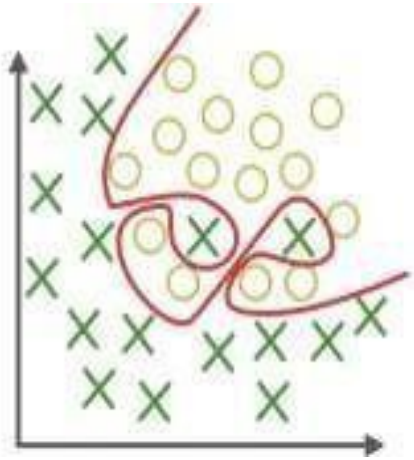
- Overfitting and Under fitting
- Bias /Variance and their tradeoffs
- Performance evaluation
- Code demo

What & Why of Overfitting and Underfitting









What & Why of Overfitting and Underfitting

Underfitting refers to a model that can neither model the training data nor generalize to new data.
An underfit machine learning model is not a suitable model and will be obvious as it will have poor performance on the training data.

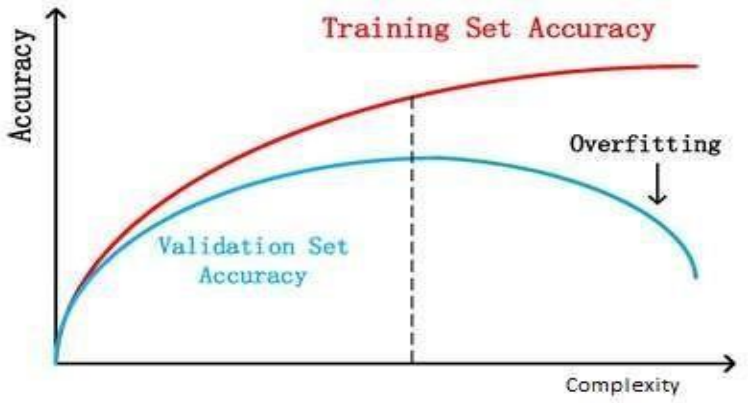
Overfitting refers to a model that models the training data too well. Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model. The problem is that these concepts do not apply to new data and negatively impact the models ability to generalize.

How to Detect Overfitting?

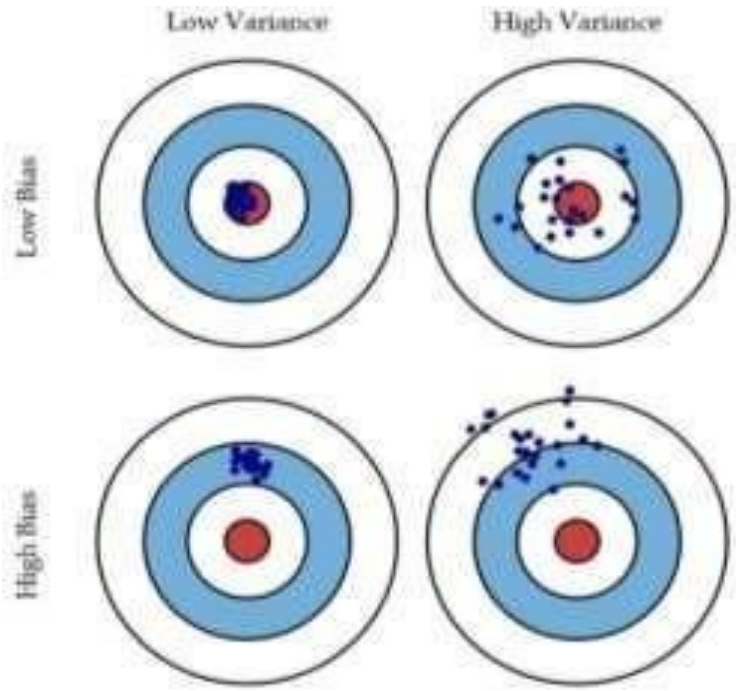
Detecting overfitting is almost impossible before you test the data. It can help address the inherent characteristic of overfitting, which is the inability to generalize data sets. The data can, therefore, be separated into different subsets to make it easy for training and testing. The data is split into three main parts, i.e., a train set ,a test set & a validate set

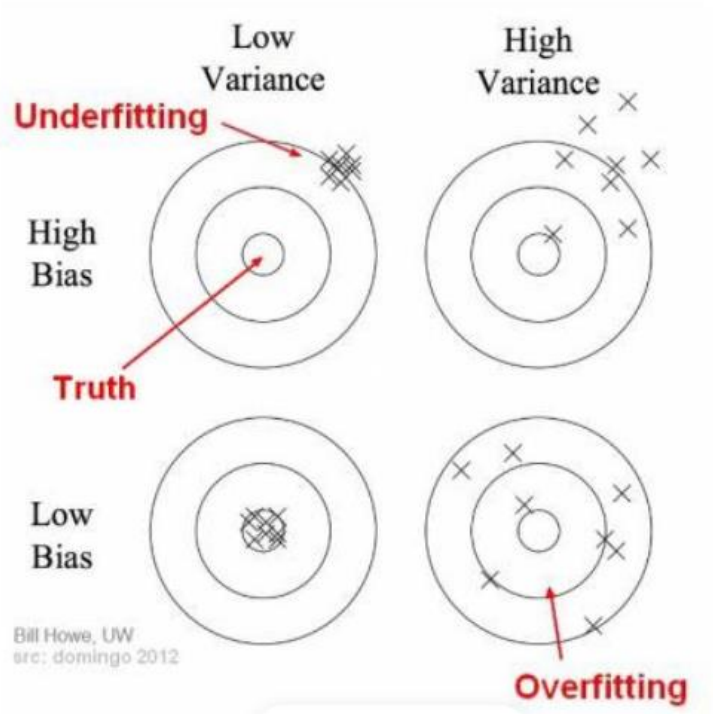
Training set: A set of examples used for learning, that is to fit the parameters of the classifier.
Validation set: A set of examples used to tune the parameters of a classifier, for example to choose the number of hidden units in a neural network.
Test set: A set of examples used only to assess the performance of a fully-specified classifier.

By segmenting the dataset, we can examine the performance of the model on each set of data to spot overfitting when it occurs. The performance can be measured using the percentage of accuracy observed in train & validate data sets to conclude on the presence of overfitting.



Bias and Variance in regression models

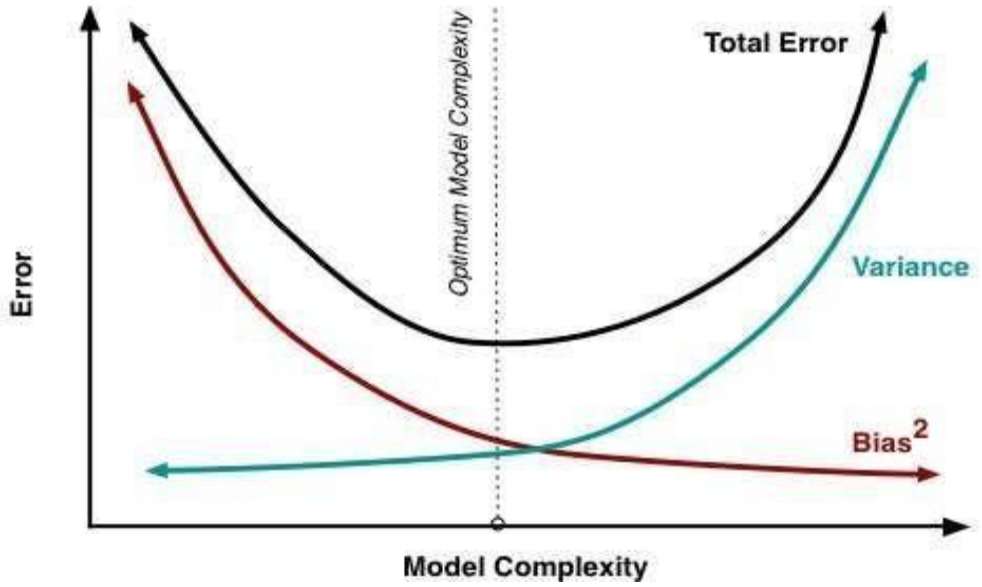




Bias-Variance Trade-Off

The goal of any supervised machine learning algorithm is to achieve low bias and low variance. In turn the algorithm should achieve good prediction performance.

- Linear machine learning algorithms often have a high bias but a low variance.
- Non-linear machine learning algorithms often have a low bias but a high variance.



The parameterization of machine learning algorithms is often a battle to balance out bias and variance.

Model Evaluation

A confusion matrix shows the number of correct and incorrect predictions made by the classification model compared to the actual outcomes (target value) in the data. The matrix is *N*x*N*, where *N* is the number of target values (classes). Performance of such models is commonly evaluated using the data in the matrix. The following table displays a 2x2 confusion matrix for two classes (Positive and Negative).

Bias and Variance in regression models

The prediction error for any machine learning algorithm can be broken down into three parts:

- Bias Error
- Variance Error
- Irreducible Error

The irreducible error cannot be reduced regardless of what algorithm is used. It is the error introduced from the chosen framing of the problem and may be caused by factors like unknown variables that influence the mapping of the input variables to the output variable.

- **Low Bias**: Suggests less assumptions about the form of the target function.
- **High-Bias**: Suggests more assumptions about the form of the target function.

- **Low Variance**: Suggests small changes to the estimate of the target function with changes to the training dataset.
- **High Variance**: Suggests large changes to the estimate of the target function with changes to the training dataset.

Regularized Linear Regression

One of the major aspects of training your machine learning model is avoiding overfitting. *The model will have a low accuracy if it is overfitting. This* happens because your model is trying too hard to capture the noise in your training dataset.

Regularization

This is a form of regression, that constrains/ regularizes or shrinks the coefficient estimates towards zero. In other words, ***this technique discourages learning a more complex or flexible model, so as to avoid the risk of overfitting.***

Lasso Regression

$$\mathrm{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 .$$

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| = \mathrm{RSS} + \lambda \sum_{j=1}^{p} |\beta_j|.$$

Ridge Regression

Residual Sum of Squares (RSS) is modified by adding the shrinkage quantity.

Now, the coefficients are estimated by minimizing this function. Here, ***λ is the tuning parameter that decides how much we want to penalize the flexibility of our model.***

The increase in flexibility of a model is represented by increase in its coefficients, and if we want to minimize the above function, then these coefficients need to be small.

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2$$

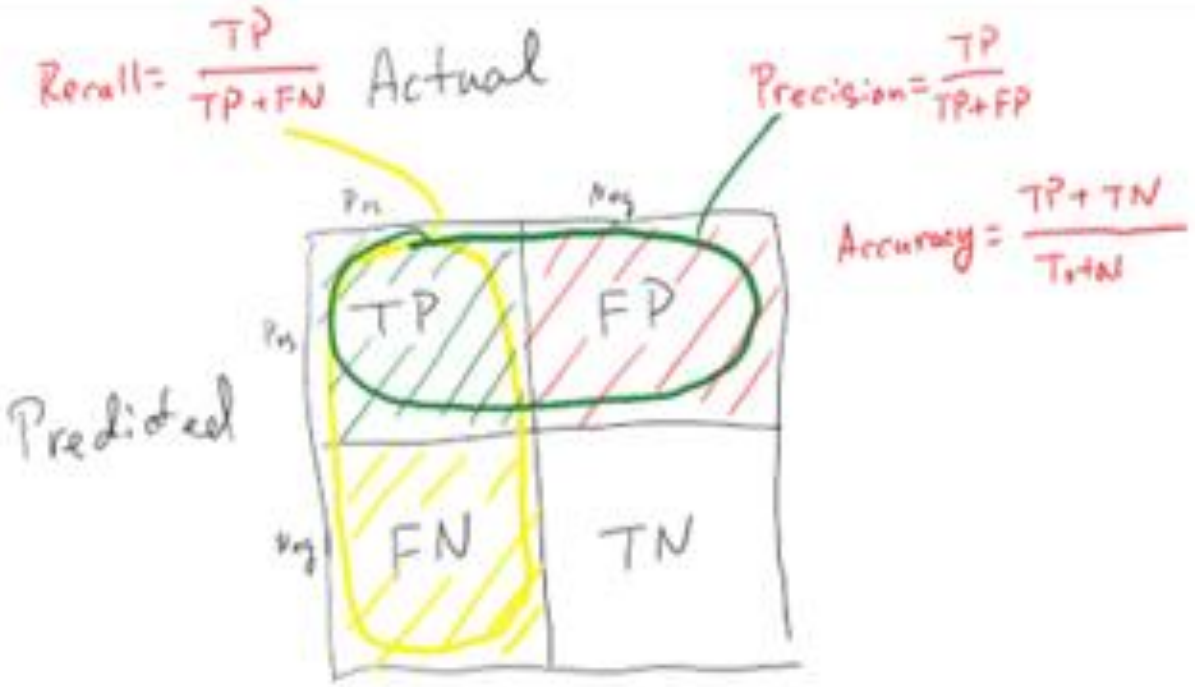
Ridge vs. Lasso Regression

- In cases where only a small number of predictor variables are significant, **lasso regression** tends to perform better because it's able to shrink insignificant variables completely to zero and remove them from the model

- when many predictor variables are significant in the model and their coefficients are roughly equal then **ridge regression** tends to perform better because it keeps all of the predictors in the mode

Model Evaluation

Predicted Values	Actual Values: Positive (1)	Actual Values: Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN



Specificity:

$$\text{Specificity} = TN/(TN+FP)$$

- A measure that evaluates the ability of a model to correctly identify the instances of the negative class.
- Specificity is complementary ro sensitivity. While sensitivity (recall) focuses on the model's ability to correctly identify **positive instances**, specificity focuses on correctly identifying **negative instances**
- Called the True Negative Rate
- Also known as the **true negative rate** or the specificity index.
- Relevant when cost of mis classification is high or when there is class imbalance
- Example: Covid vaccine:
  - specificity would refer to the ability of the test to correctly identify individuals who have not received the COVID-19 vaccine. The test aims to be highly specific to the vaccinated population, minimizing false positives among those who haven't received the vaccine.

Sensitivity/Recall

Recall = TP/(TP+FN)

- Sensitivity, also called Recall or **True Positive Rate (TPR)**, measures the proportion of actual positive instances that the model correctly identifies. It answers the question: "Of all the actual positive instances, how many did the model correctly predict as positive?"

- A high sensitivity value indicates that the model is effective at capturing most of the positive instances in the dataset.
- Sensitivity is particularly important in scenarios where the cost of missing positive instances (false negatives) is high.
- Example
  - In prediction of malignancy In medical diagnostics, identifying all cases of a disease is crucial.
  - In fraud detection, sensitivity is important to catch as many fraudulent transactions as possible, even if it results in some false positives.

False Positive Rate (FPR)

- A measure that evaluates the ability of a model to correctly identify the instances of the negative class.
- A high FPR value indicates that the model is prone to making false positive errors by incorrectly classifying negative instances as positive.
- FPR is significant when the cost or impact of making a false positive prediction is high.
- In datasets where the negative class dominates, FPR provides insights into the model's ability to avoid misclassifying negative instances as positive.
- Example: spam email classifier.
  - High FPR would mean that the model incorrectly marks a significant number of legitimate emails as spam, leading to false positives.

FPR = 1 - Specificity

$$1 - \frac{TN}{TN+FP}$$

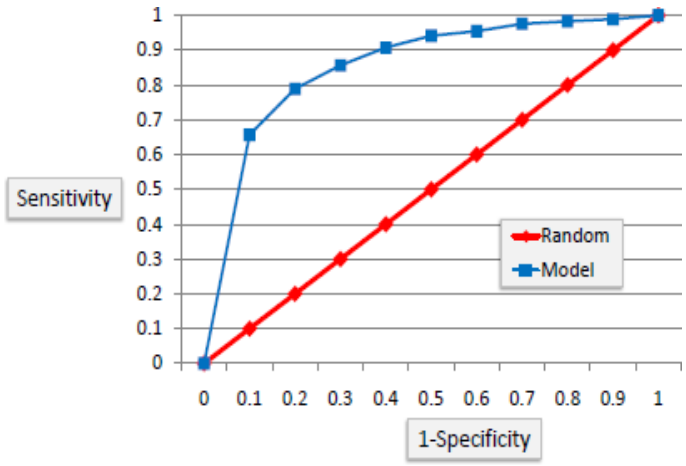
$$= \frac{FN+FP-TN}{TN+FP}$$

$$= \frac{FP}{FP+TN}$$

Model Evaluation

ROC Curves summarize the trade-off between the true positive rate and false positive rate for a predictive model using different probability thresholds

It's often used as a proxy for the trade-off between the sensitivity of the model (true positives) vs the fall- out or the probability it will trigger a false alarm (false positives).



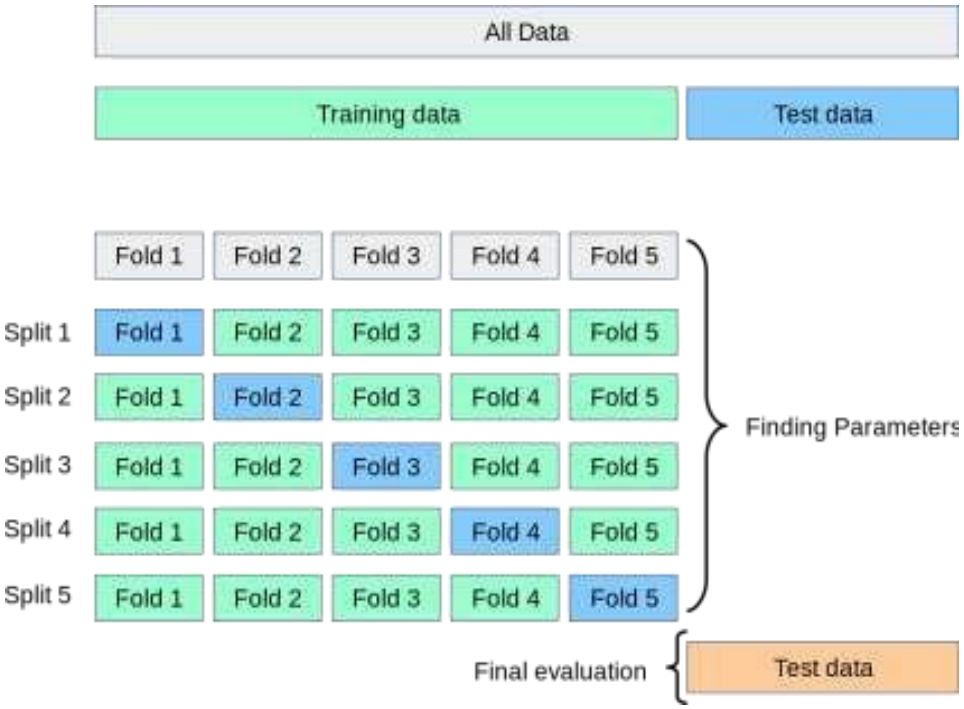
$$\text{TPR /Recall / Sensitivity} = \frac{TP}{TP + FN} \qquad \text{Specificity} = \frac{TN}{TN + FP} \qquad \textit{F - measure} = \frac{2*\text{Recall}*\text{Precision}}{\text{Recall} + \text{Precision}}$$

AUC is a measure of the discriminatory power of a model

Cross Validation

Cross-validation is a resampling technique used to evaluate machine learning models on a limited data set.

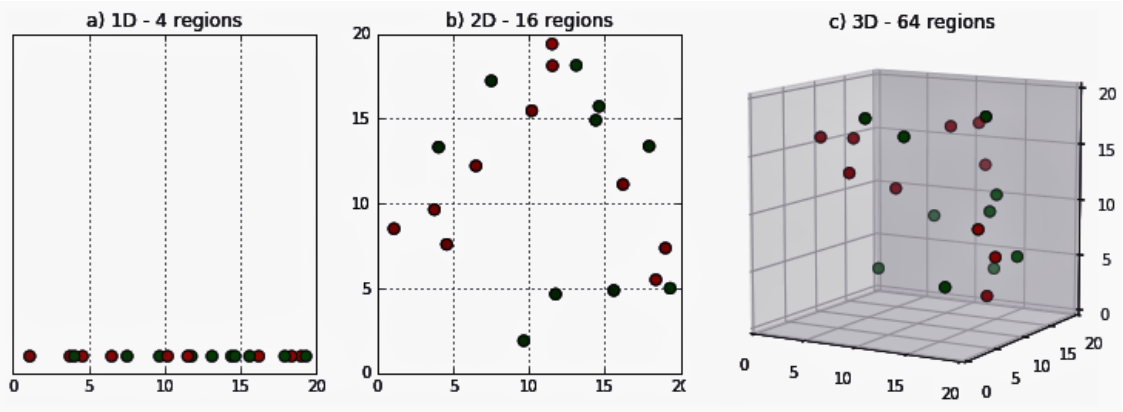
The most common use of cross-validation is the k-fold cross-validation method. Our training set is split into K partitions, the model is trained on $K-1$ partitions and the test error is predicted and computed on the Kth partition. This is repeated for each unique group and the test errors are averaged across.



Curse of dimensionality

As the number of features or dimension grows, the amount of data we need to generalize accurately grows exponentially

Sparsity of data occurs when moving to higher dimensions. the volume of the space represented grows so quickly that the data cannot keep up and thus becomes sparse



Infinite Features Requires Infinite Training

Rule of thumb for sample size

$N \geq k \times d$

where:

- N is the number of data points.
- d is the number of features.
- k is a constant (e.g. 5 or 10).

Thank You