Student ID: *015979795* ***Student Name:*** *Poojashree NS*

Train-Test Split for Evaluating Machine Learning Algorithms

The train-test split is a technique for evaluating the performance of a machine learning algorithm.

It can be used for classification or regression problems and can be used for any supervised learning algorithm.

The procedure involves taking a dataset and dividing it into two subsets. The first subset is used to fit the model and is referred to as the training dataset. The second subset is not used to train the model; instead, the input element of the dataset is provided to the model, then predictions are made and compared to the expected values. This second dataset is referred to as the test dataset.

- **Train Dataset**: Used to fit the machine learning model.
- **Test Dataset**: Used to evaluate the fit machine learning model.

The objective is to estimate the performance of the machine learning model on new data: data not used to train the model.

The train-test procedure is appropriate when there is a sufficiently large dataset available.

If you have insufficient data, then a suitable alternate model evaluation procedure would be the k-fold cross-validation procedure.

You must choose a split percentage that meets your project's objectives with considerations that include:

- Computational cost in training the model.
- Computational cost in evaluating the model.
- Training sets representativeness.
- Test set representativeness.

Latent Feature and Manifold

It's not always possible to have examples for each variable that we are trying to model. Sometimes the unobserved variables make the learning more difficult. These models tend to explain the complex relationships between several variables by drawing simple relations between the unobserved and the variables. A latent variable model p is a probability distribution over 2 sets of variables x and z

$$p(x, z; \theta)$$

Where x -> these variables are observed during learning time in Dataset z -> they are never observed. These models can be directed and undirected. Ex: Gaussian Mixture Models These are the most widely used models in machine learning.

Latent Variable Models are useful for these two reasons:

1. One reason is that some variables could be naturally unobserved. Such as if we are doing some clinical research it is very much possible that we don't have samples for few of the cases. So this model helps to work with those missing data.

2. It is also very helpful to leverage the prior knowledge we have when defining the model. This way if we already have the domain knowledge we will be able to design the model to capture those values.

Linear Factor Analysis: Since the idea behind Linear LNF is to draw the relationship between variables in X and the unobserved in latent Y plus noise, here the number of factors q is less than p. The aim is to determine the smallest value of q for which the model is adequate.

Feature Assessment and Selection

GINI Index: It is calculated by subtracting the sum of squared probabilities of each class from one. It favors larger partitions and is easy to implement whereas information gain favors smaller partitions with distinct values.

$$\text{Gini Index} = 1 - \sum_{i=1}^{n} (P_i)^2$$

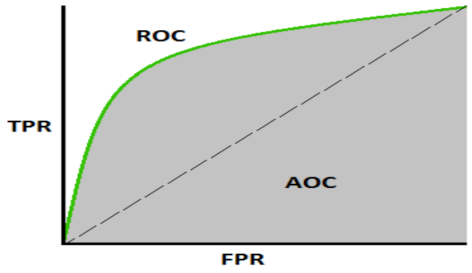
A feature with a lower Gini index is chosen for a split.
The classic CART algorithm uses the Gini Index for constructing the decision tree.

Gini Index, also known as Gini impurity, calculates the amount of probability of a specific feature that is classified incorrectly when selected randomly. If all the elements are linked with a single class then it can be called pure. The Gini index varies between values 0 and 1, where 0 expresses the purity of classification, i.e. All the elements belong to a specified class or only one class exists there. And 1 indicates the random distribution of elements across various classes. The value of 0.5 of the Gini Index shows an equal distribution of elements over some classes.

Understanding AUC - ROC Curve

AUC - ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1.

The ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis.



An excellent model has AUC near to the 1 which means it has a good measure of separability. A poor model has an AUC near 0 which means it has the worst measure of separability. In fact, it means it is reciprocating the result. It is predicting 0s as 1s and 1s as 0s. And when AUC is 0.5, it means the model has no class separation capacity whatsoever.

Confusion Matrix

As the name suggests it's a NXN matrix in which N is the number of features in the dataset or the number of classes for classification problems. It's a tabular representation of the true value and the model prediction for the N classes. Every column in the confusion matrix represents the instance value of the actual class and each row represents the instance value of the predicted class. The accuracy of the matrix can be calculated by the below formula: Accuracy = (True Positive + True Negative) / Total Sample Where True Positive means when both predicted and accurate values are the same. And True Negative means when both predicted and accurate values are different.

Lift:

Lift Score = (predicted rate) / (average rate) Lift charts help to get a better understanding of the overall performance of the model like all other evaluation methods. If the slope of the chart is non monotonic we can easily spot the flaws.