Feature Selection Techniques in Machine Learning

While building a machine learning model for real-life dataset, we come across a lot of features in the dataset and not all these features are important every time. Adding unnecessary features while training the model leads us to reduce the overall accuracy of the model, increase the complexity of the model and decrease the generalization capability of the model and makes the model biased. Even the saying "Sometimes less is better" goes as well for the machine learning model. Hence, **feature selection** is one of the important steps while building a machine learning model. Its goal is to find the best possible set of features for building a machine learning model.

Feature selection methods

For a dataset with **d features**, if we apply the hit and trial method with all possible combinations of features then total (**2^d – 1**) models need to be evaluated for a *significant* set of features. It is a time-consuming approach, therefore, we use *feature selection* techniques to find out the smallest set of features more efficiently.
There are three types of *feature selection* techniques :

1. Filter methods
2. Wrapper methods
3. Embedded methods

Filter Methods

These methods are generally used while doing the pre-processing step. These methods select features from the dataset irrespective of the use of any machine learning algorithm. In terms of computation, they are very fast and inexpensive and are very good for removing duplicated, correlated, redundant features but these methods do not remove multicollinearity. Selection of feature is evaluated individually which can sometimes help when features are in isolation (don't have a dependency on other features) but will lag when a combination of features can lead to increase in the overall performance of the model.

Set of all features → Selecting the best subset → Learning algorithm → Performance

Some techniques used are:

- **Information Gain –** It is defined as the amount of information provided by the feature for identifying the target value and measures reduction in the entropy values. Information gain of each attribute is calculated considering the target values for feature selection.
- **Chi-square test —** Chi-square method (X2) is generally used to test the relationship between categorical variables. It compares the observed values from different attributes of the dataset to its expected value.

$$X^2 = \sum \frac{(Observed\ value - Expected\ value)^2}{Expected\ value}$$

Chi-square Formula

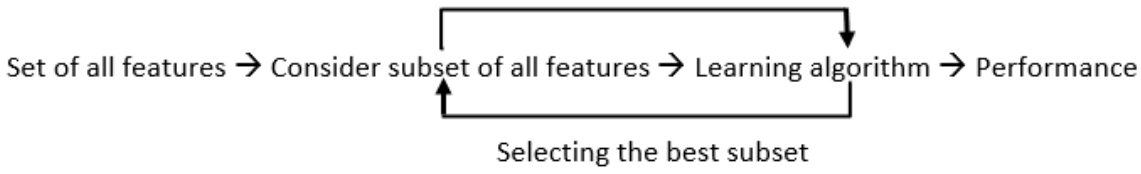
- **Fisher's Score –** Fisher's Score selects each feature independently according to their scores under Fisher criterion leading to a suboptimal set of features. The larger the Fisher's score is, the better is the selected feature.

- **Correlation Coefficient –** Pearson's Correlation Coefficient is a measure of quantifying the association between the two continuous variables and the direction of the relationship with its values ranging from *-1 to 1*.

- **Variance Threshold –** It is an approach where all features are removed whose variance doesn't meet the specific threshold. By default, this method removes features having zero variance. The assumption made using this method is higher variance features are likely to contain more information.

- **Mean Absolute Difference (MAD) –** This method is similar to variance threshold method but the difference is there is no square in MAD. This method calculates the mean absolute difference from the mean value.

Wrapper methods:
Wrapper methods, also referred to as greedy algorithms train the algorithm by using a subset of features in an iterative manner. Based on the conclusions made from training in prior to the model, addition and removal of features takes place. Stopping criteria for selecting the best subset are usually pre-defined by the person training the model such as when the performance of the model decreases or a specific number of features has been achieved. The main advantage of wrapper methods over the filter methods is that they provide an optimal set of features for training the model, thus resulting in better accuracy than the filter methods but are computationally more expensive.



Wrapper Methods Implementation

Some techniques used are:

p-value - based on the p-values, we will remove the features one by one. We will keep running the machine learning algorithm and in each iteration, we will find the feature with the highest p-value. If that highest p-value is greater than 0.05, we will remove that feature. The same process will be done till we reach a point where the highest p-value is not greater than 0.05 anymore.

Forward selection – This method is an iterative approach where we initially start with an empty set of features and keep adding a feature which best improves our model after each iteration. The stopping criterion is till the addition of a new variable does not improve the performance of the model.

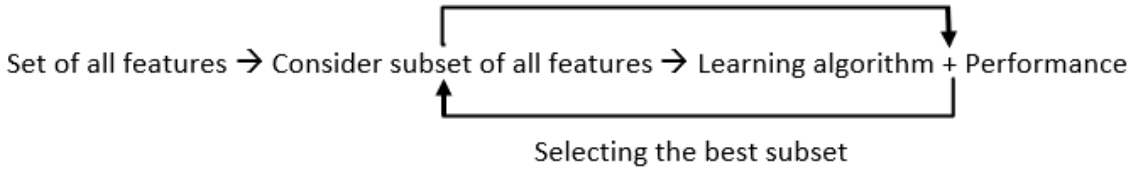
Backward elimination – This method is also an iterative approach where we initially start with all features and after each iteration, we remove the least significant feature. The stopping criterion is till no improvement in the performance of the model is observed after the feature is removed.

Stepwise: Stepwise elimination is a hybrid of forward and backward elimination and starts similarly to the forward elimination method, e.g. with no regressors. Features are then selected as described in forward feature selection, but after each step, regressors are checked for elimination as per backward elimination. The hope is that as we enter new variables that are better at explaining the dependent variable, variables already included may become redundant.

Recursive elimination – This greedy optimization method selects features by recursively considering the smaller and smaller set of features. The estimator is trained on an initial set of features and their importance is obtained using feature_importance_attribute. The least important features are then removed from the current set of features till we are left with the required number of features.

Embedded methods:

In embedded methods, the feature selection algorithm is blended as part of the learning algorithm, thus having its own built-in feature selection methods. Embedded methods encounter the drawbacks of filter and wrapper methods and merge their advantages. These methods are faster like those of filter methods and more accurate than the filter methods and take into consideration a combination of features as well.



Embedded Methods Implementation

Some techniques used are:

Regularization – This method adds a penalty to different parameters of the machine learning model to avoid over-fitting of the model. This approach of feature selection uses Lasso (L1 regularization) and

Elastic nets (L1 and L2 regularization). The penalty is applied over the coefficients, thus bringing down some coefficients to zero. The features having zero coefficient can be removed from the dataset.

Tree-based methods – These methods such as Random Forest, Gradient Boosting provides us feature importance as a way to select features as well. Feature importance tells us which features are more important in making an impact on the target feature.

Difference between Filter, Wrapper, and Embedded Methods for Feature Selection

Filter methods	Wrapper methods	Embedded methods
Generic set of methods which do not incorporate a **specific machine learning algorithm.**	Evaluates on a **specific machine learning algorithm** to find optimal features.	Embeds (fix) features during **model building process.** Feature selection is done by observing each iteration of model training phase.
Much **faster** compared to Wrapper methods in terms of time complexity	**High computation time** for a dataset with many features	Sits **between Filter methods and Wrapper methods** in terms of time complexity
Less prone to **over-fitting**	High chances of **over-fitting** because it involves training of machine learning models with different combination of features	Generally used to reduce **over-fitting** by **penalizing** the coefficients of a model being too large.
Examples – **Correlation, Chi-Square test, ANOVA, Information gain** etc.	Examples - **Forward Selection, Backward elimination, Stepwise selection** etc.	Examples - **LASSO, Elastic Net, Ridge Regression** etc.

Filter vs. Wrapper vs. Embedded methods

1. Forward selection

In *forward selection*, we start with a null model and then start fitting the model with each individual feature one at a time and select the feature with the minimum *p-value*. Now fit a model with two features by trying combinations of the earlier selected feature with all other remaining features. Again select the feature with the minimum *p-value*. Now fit a model with three features by trying combinations of two

previously selected features with other remaining features. Repeat this process until we have a set of selected features with a *p-value* of individual features less than the *significance level*.

In short, the steps for ***the forward selection*** technique are as follows :

1. Choose a *significance level* (e.g. SL = 0.05 with a 95% confidence).
2. Fit all possible *simple regression models* by considering one feature at a time. Total **'n'** models are possible. Select the feature with the lowest *p-value*.
3. Fit all possible models with one extra feature added to the previously selected feature(s).
4. Again, select the feature with a minimum *p-value*. if *p_value* < *significance level* then go to Step 3, otherwise terminate the process.

2. Backward elimination

In *backward elimination*, we start with the full model (including all the independent variables) and then remove the insignificant feature with the highest *p-value*(> *significance level*). This process repeats again and again until we have the final set of *significant* features.

In short, the steps involved in *backward elimination* are as follows:

1. Choose a *significance level* (e.g. SL = 0.05 with a 95% confidence).
2. Fit a full model including all the features.

3. Consider the feature with the highest *p-value*. If the *p-value* > *significance level* then go to Step 4, otherwise terminate the process.
4. Remove the feature which is under consideration.
5. Fit a model without this *feature*. Repeat the entire process from Step 3.