

Wrapper Methods

Wrapper methods for feature selection are a type of feature selection methods that involve using a predictive model to score the combination of features. They are called "wrapper" method because they "wrap" this type of model-based evaluation around the feature selection process.

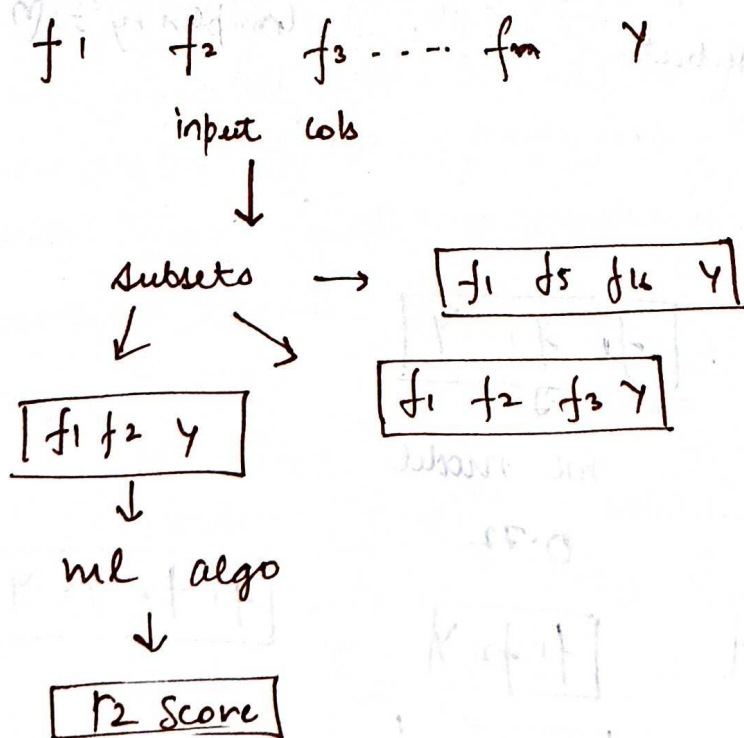
Here's how wrapper methods work in general

1. Subset Generation :- First, a subset of feature is generated. This can be done in a variety of ways. For example, you might start with one feature and gradually add more, or start with all features and gradually remove them, or generate subset generation method depends on the specific type of wrapper method being used.

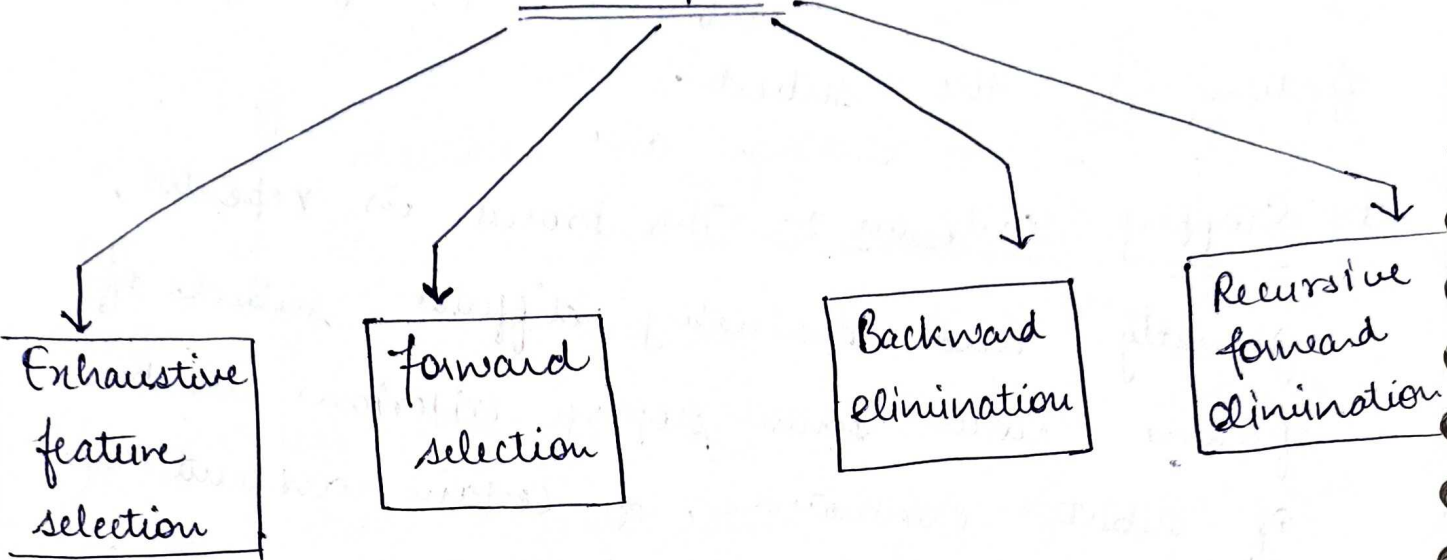
2. Subset Evaluation :- After a subset of feature has been generated, a model is trained on this subset of feature and the model's performance is evaluated usually through

Cross-validation. The performance of the model gives an estimate of the quality of the feature in the subset.

3) Stopping Criterion :- This process is repeated, generating and evaluating different subsets of features, until some stopping criterion number of subsets evaluated, a certain amount of time elapsed, no improvement in model performance after a certain number of iteration.



Wrapper



Exhaustive feature selection / Best Subset Selection

$f_1 \quad f_2 \quad f_3 \quad f_4 \quad f_5 \quad y$
 \downarrow

try out all the subset

complexity = $(2^n - 1)$

$f_1 \quad f_2 \quad y$

$\boxed{f_1 \quad y}$
 \downarrow

ml model

0.62

$\boxed{f_2 \quad y}$
 \downarrow

ml model

0.51

$\boxed{f_1 \quad f_2 \quad y}$
 \downarrow

ml model

0.72

$f_1 \quad f_2 \quad f_3$

$\rightarrow \boxed{f_1 \quad y}$

$\rightarrow \boxed{f_2 \quad y}$

$\rightarrow \boxed{f_3 \quad y}$

$\boxed{f_1 \quad f_2 \quad y}$

$\boxed{f_2 \quad f_3 \quad y}$

$\boxed{f_1 \quad f_3 \quad y}$

$\boxed{f_1 \quad f_2 \quad f_3 \quad y}$

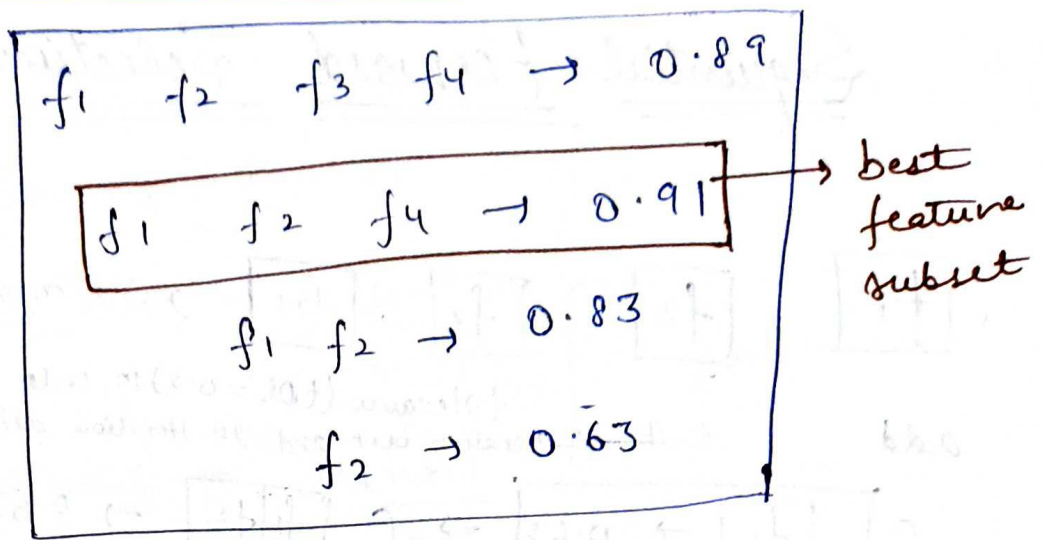
\rightarrow let this is best
 \rightarrow select $\boxed{f_1 \quad f_2}$

Disadvantage

1. Computational Complexity:- The biggest drawback is its computational cost. If you have n features, the number of combinations to check is 2^n . So as the number of features grows, the number of combinations grow exponentially, making this method computationally expensive and time consuming. For datasets with a large number of features, it may not be practical.
2. Risk of Overfitting:- By checking all possible combinations of features, there's risk of overfitting the model to the training data. The feature combination that performs best on the training data may not necessarily perform well on unseen data.
3. Require a good Evaluation Metric:- The effectiveness of exhaustive feature selection depends on the quality of the evaluation metric used to assess the goodness of a feature subset. If a poor metric is used, the feature selection may not yield optimal results.
$R^2 \rightarrow$ add feature
feature \Rightarrow adjusted R^2 square

Sequential Backward Selection / Elimination

1 iteration	$\left[\begin{array}{cccc} \boxed{f_1} & \boxed{f_2} & \boxed{f_3} & \boxed{f_4} \end{array} \rightarrow \text{model} \rightarrow 0.89 \right]$
2 iteration	$\left[\begin{array}{cccc} f_1 & \boxed{f_2} & \boxed{f_3} & \boxed{f_4} \rightarrow \text{model} \rightarrow 0.81 \\ \boxed{f_1} & f_2 & \boxed{f_3} & \boxed{f_4} \rightarrow \text{model} \rightarrow 0.71 \\ \boxed{f_1} & \boxed{f_2} & \underline{f_3} & \boxed{f_4} \rightarrow \text{model} \rightarrow 0.91 \\ \boxed{f_1} & \boxed{f_2} & \boxed{f_3} & f_4 \rightarrow \text{model} \rightarrow 0.65 \end{array} \right]$
3 iteration	$\left[\begin{array}{cccc} f_1 & \boxed{f_2} & \boxed{f_4} \rightarrow \text{model} \rightarrow 0.79 \\ \boxed{f_1} & f_2 & \boxed{f_4} \rightarrow \text{model} \rightarrow 0.81 \\ \boxed{f_1} & \boxed{f_2} & \underline{f_4} \rightarrow \text{model} \rightarrow 0.83 \end{array} \right]$
4 iteration	$\left[\begin{array}{ccc} \boxed{f_1} & \boxed{f_2} \rightarrow \text{model} \rightarrow 0.63 \\ \boxed{f_1} & f_2 \rightarrow \text{model} \rightarrow 0.53 \end{array} \right]$



$$\frac{n(n+1)}{2}$$

Disadvantage

- * At every stage removing one feature and never using it in the future what if it was important.

Sequential Forward Selection

f_1 f_2 f_3 $f_4 \rightarrow 0 \rightarrow \text{model} + y \text{ mean}$

add

* if 1st iteration best and 2nd iteration difference

tolerance (tol = 0.5) in code

at least 0.5
if not
then kill
the process

$f_1 \rightarrow 0.63$

$f_2 \rightarrow 0.11$

$f_3 \rightarrow 0.43$

$f_4 \rightarrow 0.49$

$f_1 f_2 \rightarrow 0.63$

$f_1 f_3 \rightarrow 0.71$

$f_1 f_4 \rightarrow 0.80$

$\frac{1}{2}$

$f_1 f_2 f_4 \rightarrow 0.81$

$f_1 f_4 f_3 \rightarrow 0.85$

$f_1 f_4 f_3 f_2 = 0.83$

$f_1 \rightarrow 0.63$

$f_1 f_4 \rightarrow 0.80$

$\rightarrow f_1 f_4 f_3 \rightarrow 0.85$

$f_1 f_4 f_3 f_2 \rightarrow 0.83$

$f_1 f_4 f_3 \rightarrow \text{model}$

1 iteration $\rightarrow 4$

2 " $\rightarrow 3$

3 " $\rightarrow 2$

4 " $\rightarrow 1$

1 $\rightarrow n$

2 $\rightarrow n-1$

3 $\rightarrow n-2$

4 $\rightarrow n-3$

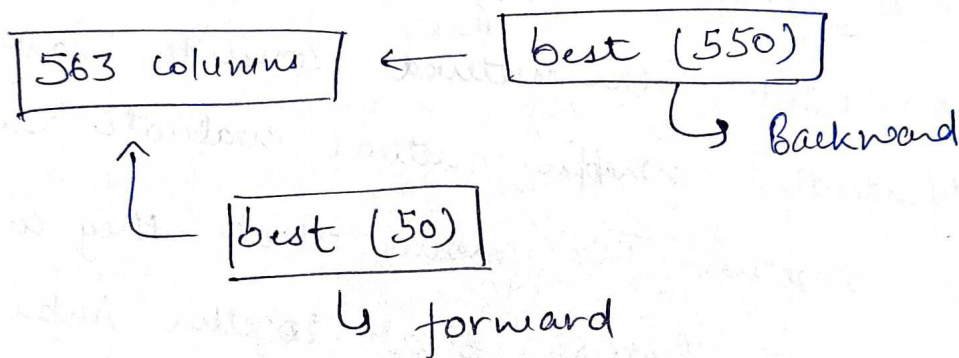
$$\frac{n(n+1)}{2}$$

n - features

$$\left\{ \begin{array}{l} \text{Exhaustive} \rightarrow 2^n - 1 \\ \text{Backward} \rightarrow \frac{n(n+1)}{2} \\ \text{Forward} \rightarrow \frac{n(n+1)}{2} \end{array} \right\}$$

* when use? ↓

Backward $\overset{\text{or}}{\leftrightarrow}$ forward



If we want overall best than anyone method use (Backward or forward)

If we want specific best then depend on method.

Disadvantage

1) Use local best not use global best and global combination.

Advantage and Disadvantage

Advantages

1. Accuracy :- Wrapper method usually provide the best performing feature subset for a given a given machine learning algorithm itself for feature selection.
2. Interaction of Feature :- They consider the interaction of feature. While filter method consider each filter independently, wrapper method evaluate subsets of feature together. This means that they can find groups of feature that together improve the performance of the model, even if individually these features are not strong predictors.

Disadvantage

1. Computational Complexity :- The main downside of wrapper method is their computation cost. As they work by generating and evaluating many diff. subset of features, they can be very time consuming, especially for datasets with a large no. of features.

2. Risk of Overfitting:- Because wrapper methods optimize the feature subset to maximum the performance of a specific machine learning model, they might select feature subset that performs well on the training data but not as well on unseen data, leading to overfitting.

3. Model specification:- The selected feature subset is tailored to maximum the performance of the specific model used in the feature selection process. Therefore, this subset might not perform as well with a different type of model.