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) <u>Stopping criterion</u> :- 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.

$f_1 \quad f_2 \quad f_3 \; \dots \; f_m \quad Y$

input cols

↓

subsets → $\boxed{f_1 \; f_5 \; f_{16} \; Y}$

↙ ↘

$\boxed{f_1 \; f_2 \; Y}$ $\quad$ $\boxed{f_1 \; f_2 \; f_3 \; Y}$

↓

ml algo

↓

$\boxed{r2 \text{ Score}}$

Wrapper

- Exhaustive feature selection
- Forward selection
- Backward elimination
- Recursive forward elimination

Exhaustive feature selection / Best Subset Selection

$f_1 \quad f_2 \quad f_3 \quad f_4 \quad f_5 \quad Y$

↓

try out all the subset

complexity = $(n^2 - 1)$

$f_1 \quad f_2 \quad Y$

$f_1 \; Y$	$f_2 \; Y$	$f_1 \; f_2 \; Y$
↓	↓	↓
ml model	ml model	ml model
0·62	0·51	0·72

$f_1 \quad f_2 \quad f_3$ → $\boxed{f_1 \; Y}$ → $\boxed{f_2 \; Y}$ → $\boxed{f_3 \; Y}$

$\boxed{f_1 \; f_2 \; Y}$

$\boxed{f_2 \; f_3 \; Y}$

$\boxed{f_1 \; f_3 \; Y}$ → let this is best

→ select $\boxed{f_1 \; f_2}$

$\boxed{f_1 \; f_2 \; f_3 \; Y}$

Disadvantage

1. <u>Computational Complexity</u> :- The biggest drawback is its computational cost. If you have n features, the number of combination 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 feature, it may not be practical.

2. <u>Risk of Overfitting</u> :- By checking all possible combinations of feature, there's risk of overfitting the model to the training data. The feature combination the perform best on the training data may not neccessarily perform well on unseen data.

3. <u>Require a Good Evaluation Metric</u> :- 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.

$\boxed{r2}$ → add feature

feature ↑↑ ⇒ use $\boxed{\text{adjusted } r2 \text{ square}}$

Sequential Backward Selection / Elimination

1 iteration

$\boxed{f_1}$ $\boxed{f_2}$ $\boxed{f_3}$ $\boxed{f_4}$ → model → 0.89

2 iteration

f_1 $\boxed{f_2}$ $\boxed{f_3}$ $\boxed{f_4}$ → model → 0.81

$\boxed{f_1}$ f_2 $\boxed{f_3}$ $\boxed{f_4}$ → model → 0.71

$\boxed{f_1}$ $\boxed{f_2}$ f_3 $\boxed{f_4}$ → model → 0.91

$\boxed{f_1}$ $\boxed{f_2}$ $\boxed{f_3}$ f_4 → model → 0.65

3 iteration

f_1 $\boxed{f_2}$ $\boxed{f_4}$ → model → 0.79

$\boxed{f_1}$ f_2 $\boxed{f_4}$ → model → 0.81

$\boxed{f_1}$ $\boxed{f_2}$ f_4 → model → 0.83

4 iteration

f_1 $\boxed{f_2}$ → model → 0.63

$\boxed{f_1}$ f_2 → model → 0.53

$f_1 \quad f_2 \quad f_3 \quad f_4 \rightarrow 0.89$

$\boxed{f_1 \quad f_2 \quad f_4 \rightarrow 0.91}$ → best feature subset

$f_1 \quad f_2 \rightarrow 0.83$

$f_2 \rightarrow 0.63$

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

Disadvantage

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

Sequential Forward Selection

f1 f2 f3 f4 → 0 → model → y mean

tolerance (tol = 0.5) in code

* if 1st iteration best and 2nd iteration differance atleast 0.5 if not then kill the process

add

f1 → 0.63 ⇒ f1 f2 → 0.63

f2 → 0.71   f1 f3 → 0.71

f3 → 0.43   f1 f4 → 0.80

f4 → 0.49

⇓

f1 f2 f4 → 0.81

f1 f4 f3 → 0.85

{ f1 f4 f3 f2 = 0.83

f1 f4 f3 → model

f1 → 0.63

f1 f4 → 0.80

→ f1 f4 f3 → 0.85

f1 f4 f3 f2 → 0.83

1 iteration → 4

2 " → 3

3 " → 2

4 " → 1

1 → n

2 → n-1

3 → n-2

4 → n-3

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

n - features

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

* when use?]

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

$\boxed{\text{563 columns}}$ ← $\boxed{\text{best (550)}}$ ↳ Backward

↑ $\boxed{\text{best (50)}}$ ↳ forward

If we want overall best than anyone methode 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 selects 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 perforomance of the specific model used in the feature selection process. Therefore, this subset might not perform as well with a different type of model.