Machine Learning Model Pipeline
Feature Selection

Why Do We Select Features?

- Simple models are easier to interpret
- Shorter training times
- Enhanced generalisation by reducing overfitting
- Easier to implement by software developers → Model production
- Reduced risk of data errors during model use
- Data redundancy

Reducing features for model deployment

- Smaller json messages sent over to the model
    - Json messages contain only the necessary variables / inputs
- Less lines of code for error handling
    - Error handlers need to be written for each variable / input
- Less information to log
- Less feature engineering code

Variable Redundancy

Constant variables

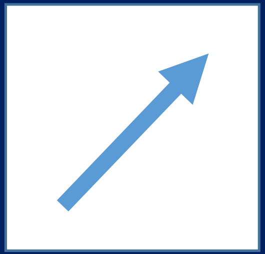
Only 1 value per variable

Quasi – constant Variables

> 99% of observations show same value

Duplication

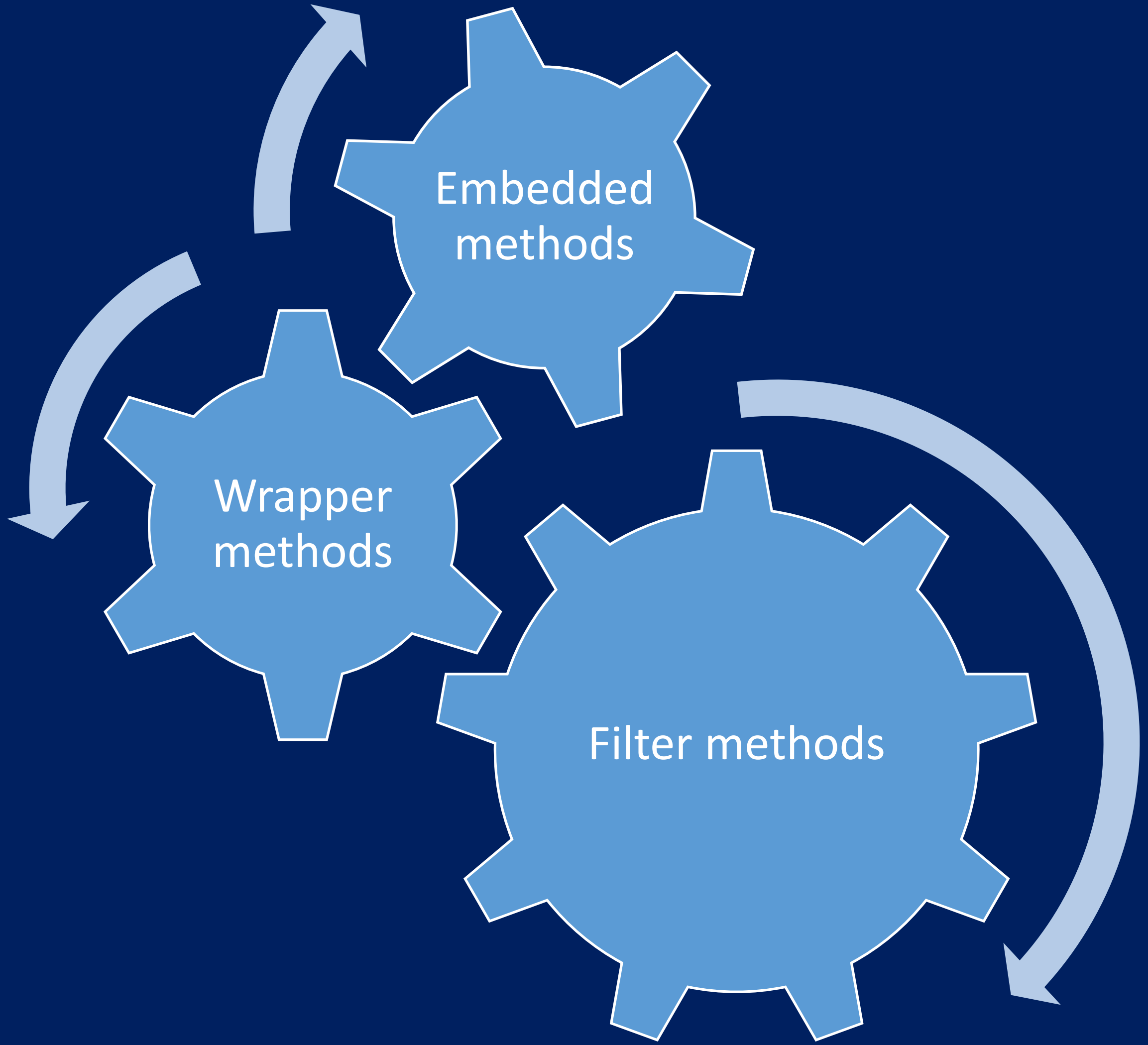
Same variable multiple times in the dataset



Correlation

Correlated variables provide the same information

Feature Selection Methods

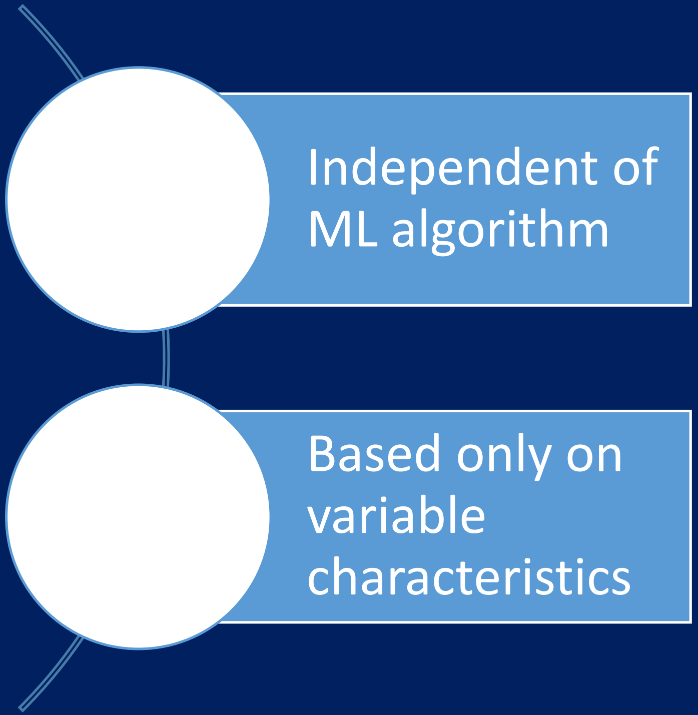


Embedded methods

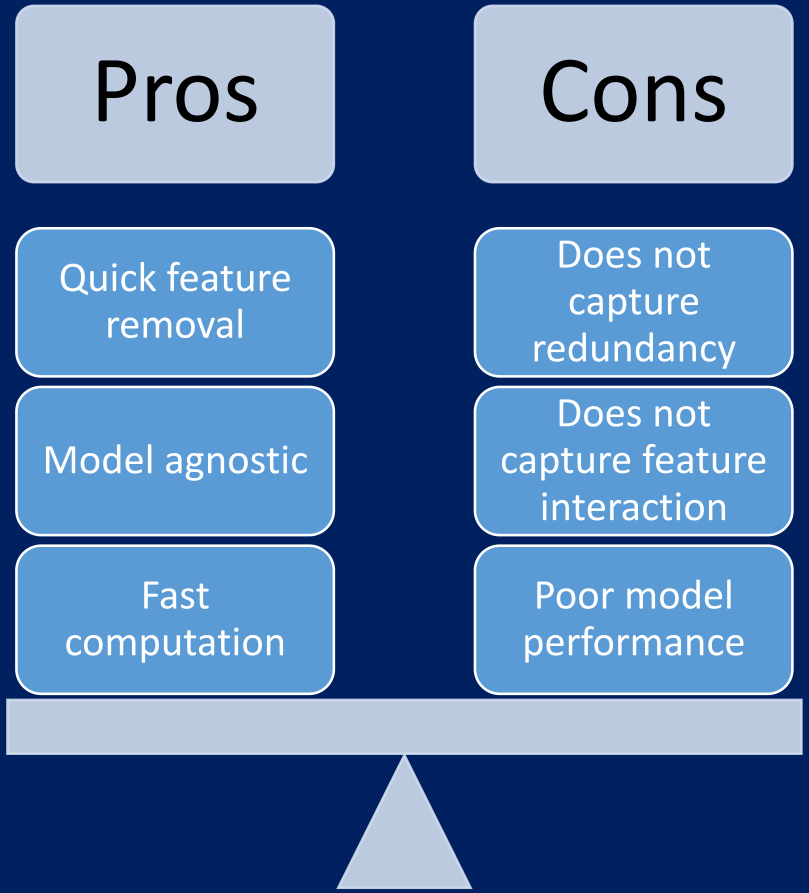
Wrapper methods

Filter methods

Filter methods



- Independent of ML algorithm
- Based only on variable characteristics



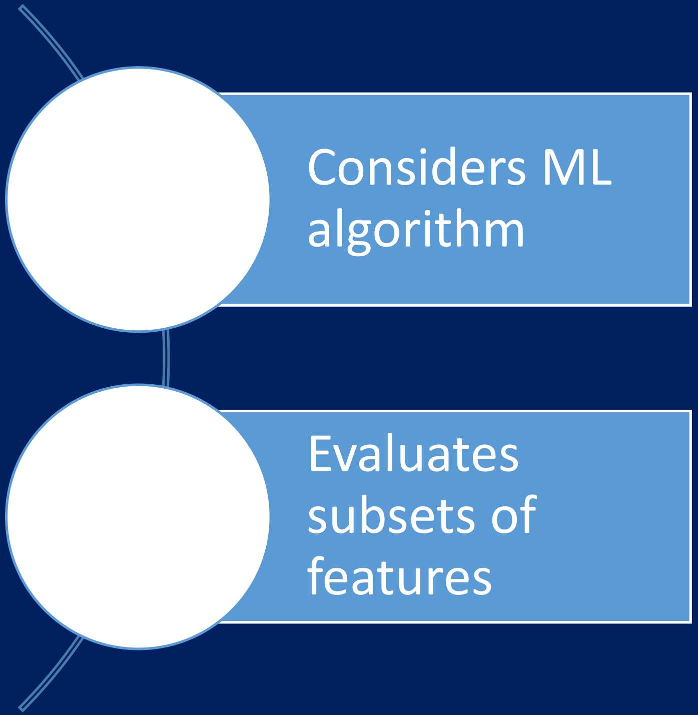
Pros

- Quick feature removal
- Model agnostic
- Fast computation

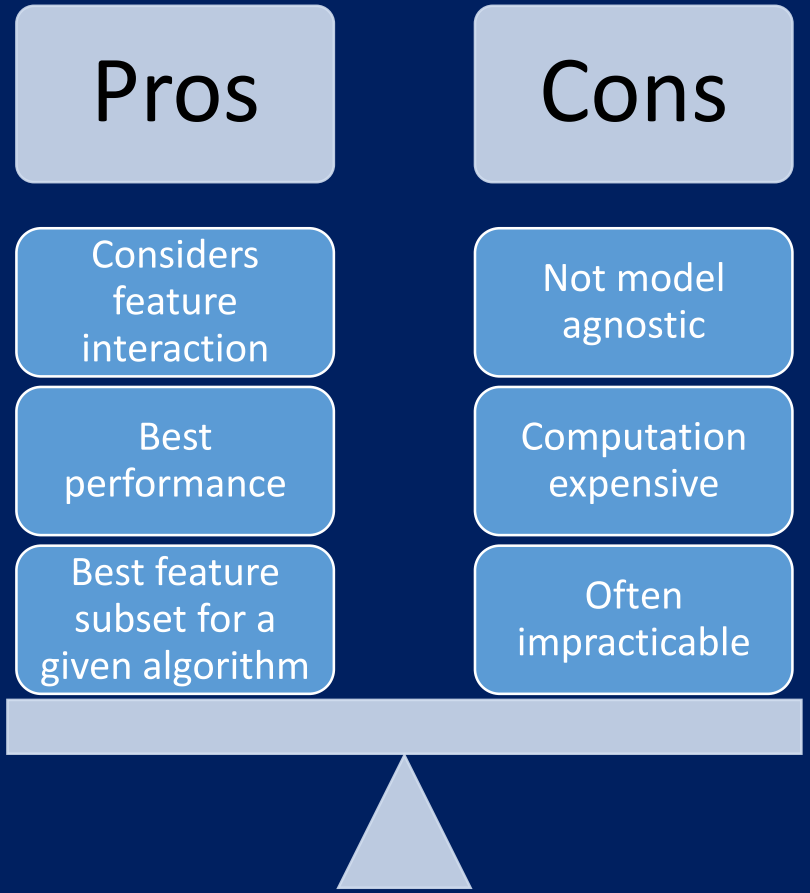
Cons

- Does not capture redundancy
- Does not capture feature interaction
- Poor model performance

Wrapper methods

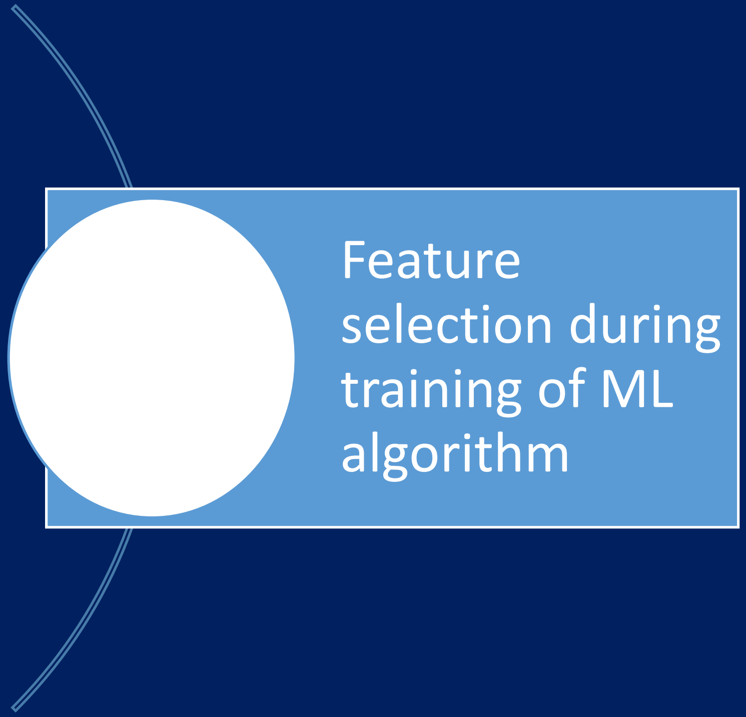


- Considers ML algorithm
- Evaluates subsets of features

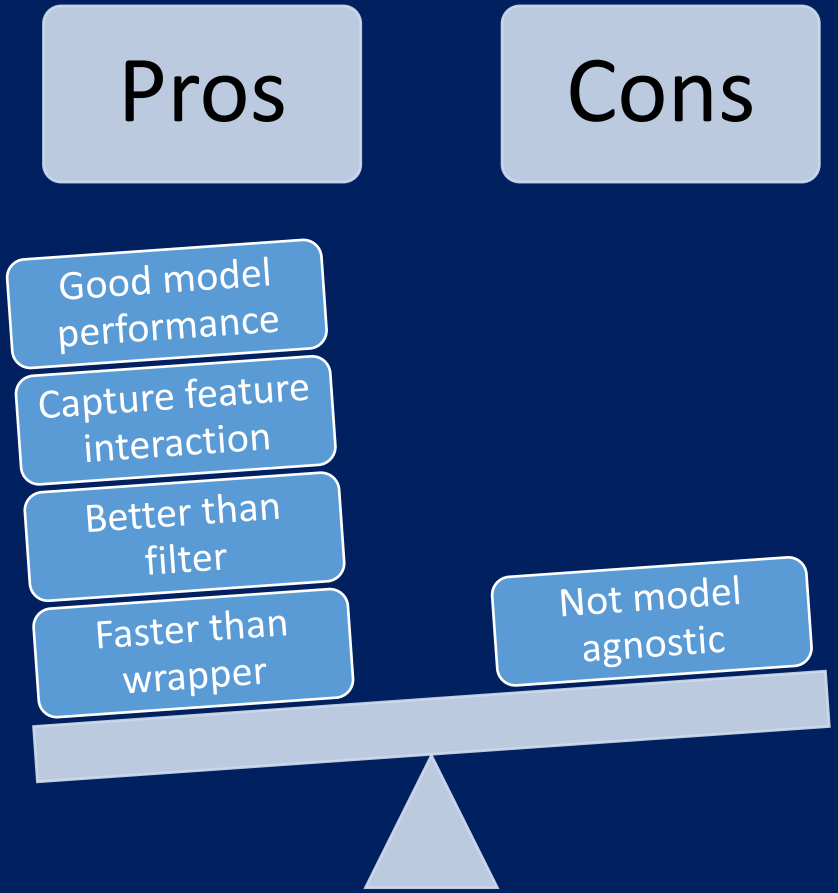


Pros	Cons
Considers feature interaction	Not model agnostic
Best performance	Computation expensive
Best feature subset for a given algorithm	Often impracticable

Embedded methods



Feature selection during training of ML algorithm



Pros
Cons
Good model performance
Capture feature interaction
Better than filter
Faster than wrapper
Not model agnostic