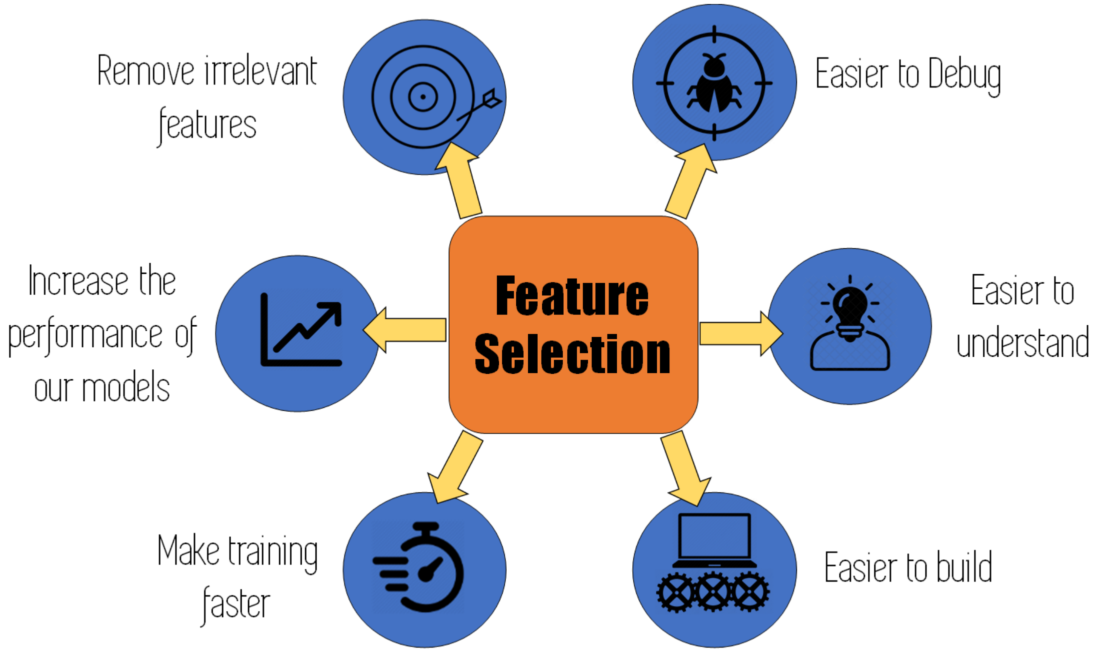
📍

Feature Selection :

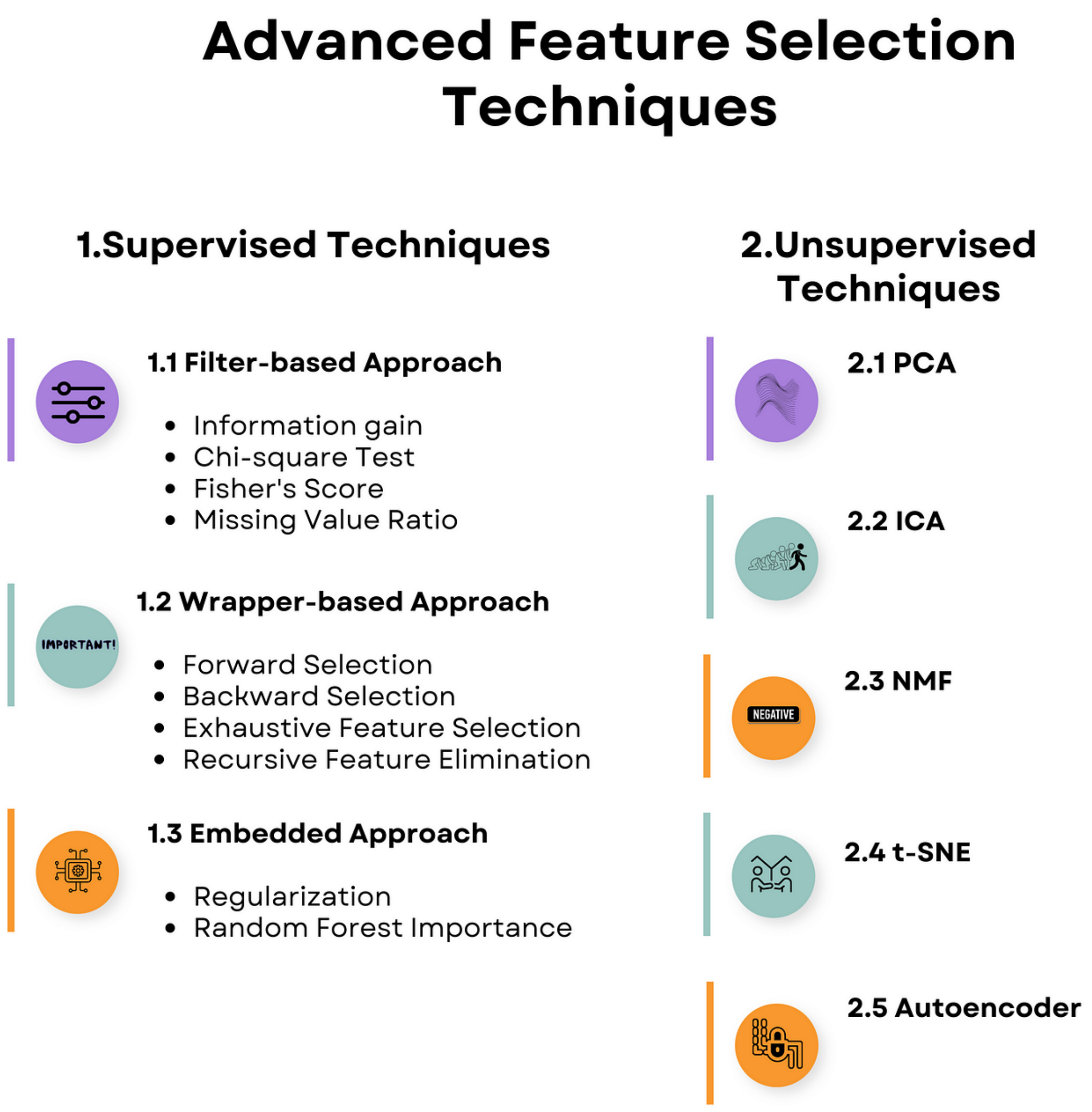
Feature selection is a process that chooses a subset of features from the original features so that the feature space is optimally reduced according to a certain criterion .

The role of feature selection in machine learning is :

1. To reduce the dimensionality of feature space.
2. To speed up a learning algorithm.
3. To improve the predictive accuracy of a classification algorithm.
4. To improve the comprehensibility of the learning results.
5.



THE MOST POPLULAR METHODS :



Advanced Feature Selection Techniques

1.Supervised Techniques

1.1 Filter-based Approach

- Information gain
- Chi-square Test
- Fisher's Score
- Missing Value Ratio

1.2 Wrapper-based Approach

- Forward Selection
- Backward Selection
- Exhaustive Feature Selection
- Recursive Feature Elimination

1.3 Embedded Approach

- Regularization
- Random Forest Importance

2.Unsupervised Techniques

2.1 PCA

2.2 ICA

2.3 NMF

2.4 t-SNE

2.5 Autoencoder

Filter methods :

Filter methods rank features based on their statistical properties and select the top-ranked features.

Filter methods are the simplest and most computationally efficient methods for feature selection. In this approach, features are selected based on their statistical properties, such as their correlation with the target variable or their variance. These methods are easy to implement and are suitable for datasets with a large number of features. However, they may not always produce the best results as they do not take into account the interactions between features.

Removing features with low variance:

`VarianceThreshold` is a simple baseline approach to feature selection. It removes all features whose variance doesn’t meet some threshold. By default, it removes all zero-variance features, i.e. features that have the same value in all samples.

As an example, suppose that we have a dataset with boolean features, and we want to remove all features that are either one or zero (on or off) in more than 80% of the samples. Boolean features are Bernoulli random variables, and the variance of such variables is given by

$$\mathrm{Var}[X] = p(1 - p)$$

Code :

```
from sklearn.feature_selection import VarianceThreshold
X = [[0, 0, 1], [0, 1, 0], [1, 0, 0], [0, 1, 1], [0, 1, 0], [0, 1, 1]]
sel = VarianceThreshold(threshold=(.8 * (1 - .8)))
sel.fit_transform(X)
```

Result :

```
array([[0, 1],
       [1, 0],
       [0, 0],
        [1, 1],
        [1, 0],
        [1, 1]])
```

Univariate feature selection :

Univariate feature selection works by selecting the best features based on univariate statistical tests. It can be seen as a preprocessing step to an estimator. Scikit-learn exposes feature selection routines as objects that implement the `transform` method:

- `SelectKBest` removes all but the K highest scoring features
- `SelectPercentile` removes all but a user-specified highest scoring percentage of features
- using common univariate statistical tests for each feature: false positive rate `SelectFpr`, false discovery rate `SelectFdr`, or family wise error `SelectFwe`.
- `GenericUnivariateSelect` allows to perform univariate feature selection with a configurable strategy. This allows to select the best univariate selection strategy with hyper-parameter search estimator.

```
from sklearn.datasets import load_iris
from sklearn.feature_selection import SelectKBest, chi2, f_class

features, labels = load_iris(return_X_y=True)

print(features.shape)
mutual_info_classif(features, labels)
# features_new = SelectKBest(f_classif, k=1).fit_transform(featu
# features_new.shape
```

Ext Feature selection :