**What is Feature Selection?**

In machine learning and data science in general, Feature selection (also known as variable selection, attribute selection or subset selection) is the process by which a data scientist selects automatically or manually a subset of relevant features to use in machine learning model building.

In fact, it is one of the core concepts in machine learning which has a huge impact on your models’ performance, as it is the key to make reliable machine learning models.

Given a pool of features, the process will select the best subset of attributes that are most important and have high contribution at the time of prediction making.

Machine learning models are moving closer and closer to edge devices. Fritz AI is here to help with this transition. [Explore our suite of developer tools that makes it easy to teach devices to see, hear, sense, and think.](https://www.fritz.ai/product/platform.html?utm_campaign=buildmodels4&utm_source=heartbeat)

**Why should we select features?**

It’s not always true that the more data features you have, the better the resulting model is going to be.

Data sometimes include irrelevant features that don’t result in better predictions, and sometimes redundant features that are irrelevant in the presence of others will make the learning process difficult and can result in overfitting.

Therefore, we need some techniques that will eliminate any attribute that’s could adversely affect learning.

There are several more reasons to complete feature selection, such as:

* **Simple models are easier to interpret:** It’s much easier to understand the output of a model that uses 10 variables than an output that uses 100 variables.
* **Shorter training time:** Reducing the number of variables reduces the computation cost, speeds up model training, and perhaps most importantly—simpler models tend to have faster prediction times.
* **Enhanced generalization by reducing overfitting:** Oftentimes, many of the variables are just noise with little prediction value. However, the ML model learns from this noise and causes overfitting while simultaneously reducing generalization. By eliminating these irrelevant noisy features, we can substantially improve the generalization of ML models.
* **Variable redundancy:** Features of a given dataset are frequently highly correlated, and we know that highly-correlated features provide the same information, which makes them redundant. In cases like these, we can keep just one feature and remove the redundant features without losing any information. Less redundant data means less opportunity for the model to make noise-based predictions.

**Feature Selection vs. Feature Engineering**

Often, newcomers to the field of machine learning may get confused between feature selection and feature engineering.

[Feature engineering](https://heartbeat.fritz.ai/hands-on-with-feature-engineering-techniques-broad-introduction-def389c1fc25) allows us to **create new features** from the ones they already have in order to help the machine learning model make more effective and accurate predictions.

Feature selection, on the other hand, allows us to **select features** from the feature pool (including any newly-engineered ones) that will help machine learning models more efficiently make predictions on target variables.

In a typical ML pipeline, we perform feature selection after completing feature engineering.

**Feature Selection vs. Dimensionality Reduction**

Dimensionality reduction is another concept that newcomers tend to lump together with feature selection. While **dimensionality reduction** uses unsupervised algorithms to reduce the number of features in a dataset (as feature selection methods do), there is an important difference:

* Feature selection is basically a process that selects and excludes some features **without modifying** them at all.
* Dimensionality reduction **modifies or transforms** features into a lower dimension. In essence, dimensionality reduction creates a whole new feature space that looks approximately like the first one, but smaller in terms of dimensions.

**The Procedure of Feature Selection**

Feature selection can be described in two steps:

* Combination of a search technique for proposing a new feature subset.
* An evaluation measure that scores how well is the different feature subsets.

This can be computationally expensive. From all the available features, we’re looking for the best combination of feature subsets. It can be more challenging because different feature subsets render optimal performances for different machine learning algorithms.

Machine learning is rapidly moving closer to where data is collected — edge devices. [Subscribe to the Fritz AI Newsletter to learn more about this transition and how it can help scale your business.](https://www.fritz.ai/newsletter?utm_campaign=fritzai-newsletter-scale6&utm_source=heartbeat)

**Feature selection methods**

Generally speaking, feature selection methods can be divided into three main categories:

* **Filter Methods:** Rely on the features’ characteristics without using any machine learning algorithm. Very well-suited for a quick “screen and removal” of irrelevant features.
* **Wrapper methods:** Consider the selection of a set of features as a search problem, then uses a predictive machine learning algorithm to select the best feature subset. In essence, these methods train a new model on each feature subset, which makes it obviously very computationally expensive. However, they provide the best performing feature subset for a given machine learning algorithm.
* **Embedded methods:**Just like the wrapper methods, embedded methods take the interaction of features and models into consideration. They also perform feature selection as part of the model construction process, and they are less computationally expensive.

At the end of the series, we’ll explore more advanced techniques that use deep learning, heuristic searches, and several other ways to select features.

We’ll cover all of these methods in this series with a hands-on approach, using Python and many of its well-known data science libraries.