Filter Methods: Definition

Filter methods select features from a dataset independently for any machine learning algorithm. These methods rely only on the characteristics of these variables, so features are filtered out of the data before learning begins.

These methods are powerful and simple and help to quickly remove features— and they are generally the first step in any feature selection pipeline.

Filter Methods: Advantages

1. Selected features can be used in any machine learning algorithm,
2. They’re computationally inexpensive—you can process thousands of features in a matter of seconds.

Filter methods are very good for eliminating irrelevant, redundant, constant, duplicated, and correlated features.

Filter Methods: Types

There are two types of filter methods: **Univariate**and **Multivariate.**

**Univariate filter methods** evaluate and rank a single feature according to certain criteria.

They treat each feature individually and independently of the feature space. This is how it functions in practice:

* It ranks features according to certain criteria.
* Then select the highest ranking features according to those criteria.

One problem that can occur with univariate methods is they may select a redundant variable, as they don’t take into consideration the relationship between features.

**Multivariate filter methods**, on the other hand, evaluate the entire feature space. They take into account features in relation to other ones in the dataset.

These methods are able to handle duplicated, redundant, and correlated features.

The Methods

In recent years, numerous methods and techniques have been proposed for univariate and multivariate filter-based feature selection. In the remainder of this article, we’ll explore the following methods in-depth, along with some code:

* Basic Filter Methods
* Correlation Filter Methods
* Statistical & Ranking Filter Methods

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)

Basic Filter Methods

These basic and intuitive methods help to remove:

* **Constant Features** (**VarianceThreshold)** that show single values in all the observations in the dataset. These features provide no information that allows ML models to predict the target.
* **Quasi-Constant** **Features** in which a value occupies the majority of the records.

Refer gethib for code.

# ****Correlation Filter Methods****

Besides duplicate features, a dataset can also include correlated features.

**Correlation**is defined as a measure of the **linear**relationship between two quantitative variables, like height and weight. You could also define correlationis a measure of how strongly one variable depends on another.

A high correlation is often a useful property—if two variables are highly correlated, we can predict one from the other. Therefore, we generally look for features that are highly correlated with the target, especially for linear machine learning models.

However, if two variables are highly correlated among themselves, they provide redundant information in regards to the target. Essentially, we can make an accurate prediction on the target with just one of the redundant variables.

In these cases, the second variable doesn’t add additional information, so removing it can help to reduce the dimensionality and also the added noise.

There are a number of methods to measure the correlation between variables—let’s explore the most widely used ones.

## **Pearson correlation coefficient**

This is a popular measure we use frequently in machine learning. It’s used to summarize the strength of the linear relationship between two data variables, which can vary between 1 and -1:

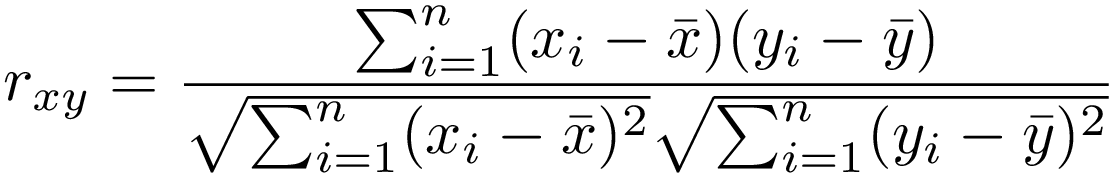
* 1 means a positive correlation: the values of one variable increase as the values of another increase.
* -1 means a negative correlation: the values of one variable decrease as the values of another increase.
* 0 means no linear correlation between the two variables.

The assumptions that the Pearson correlation coefficient makes:

* Both variables should be normally distributed.
* A straight-line relationship between the two variables.
* Data is equally distributed around the regression line.

The following formula is used to calculate the value of the Pearson correlation coefficient:

Image for post



## **Spearman’s rank correlation coefficient**

Sometimes two variables can be related in a nonlinear relationship, which can be stronger or weaker across the distribution of the variables.

And this is where Spearman’s rank correlation coefficient comes into play. It’s a non-parametric test that’s used to measure the degree of association between two variables with a monotonic function, meaning an increasing or decreasing relationship.

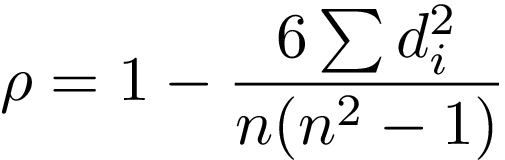
The measured strength between the variables using Spearman’s correlation varies between+1 and −1, which occurs when each of the variables is a perfect monotone function of the other. It's a lot like Pearson’s correlation, but whereas Pearson’s correlation assesses linear relationships, Spearman’s correlation assesses monotonic relationships (whether linear or not).

Spearman’s coefficient is suitablefor both continuous and discrete ordinal variables.

The Spearman’s rank correlation test doesn’t carry any assumptions about the distribution of the data.

The following formula is used to calculate the value of Spearman’s rank correlation coefficient:

Image for post

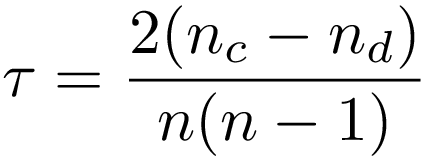


## **Kendall’s rank correlation coefficient**

Kendall’s rank correlation coefficient is a non-parametric test that measures the strength of the ordinal association between two variables. It calculates a normalized score for the number of matching or concordant rankings between the two data samples.

Kendall’s correlation varies between 1 and -1. It will take the value of 1 (high) when observations have a similar rank between the two variables, and value of -1 (low) when observations have a dissimilar rank between the two variables.

This type of correlation is best suitedfor discrete data. The following formula is used to calculate the value of Kendall’s rank correlation:



Refer Github for code.

The corr() function takes a parameter called method, which allows us to specify the type of correlation. By default it is the pearson correlation, but it can also be specified as kendall or spearman.

# Statistical & Ranking Filter Methods

These methods are statistical tests that evaluate each feature individually. By shedding light on the target, they evaluate whether the variable is important in order to discriminate against the target.

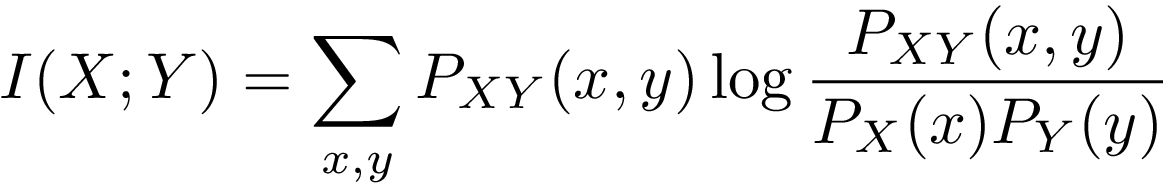
Essentially, these methods rank the features based on certain criteria or metrics and then select the features with the highest ranking.

## **Mutual Information**

Mutual information a measure of the mutual dependence of two variables. It measures the amount of information obtained about one variable through observing the other variable. In other words, it determines how much we can know about one variable by understanding another—it’s a little bit like correlation, but mutual information is more general.

In machine learning, mutual information measures how much information the presence/absence of a feature contributes to making the correct prediction on Y.

If X and Y are independent, their MI is Zero. If X is deterministic of Y, then MI is the entropy of X, which is a notion in information theory that measures or quantifies the amount of information within a variable.



## **Chi-squared Score**

This is another statistical method that’s commonly used for testing relationships between categorical variables.

Therefore, it’s suited for categorical variables and binary targets only, and the variables should be non-negative and typically boolean, frequencies, or counts.

What it does is simply compare the observed distribution between various features in the dataset and the target variable.

Here’s a code sample:

## **ANOVA Univariate Test**

A univariate test, or more specifically ANOVA ( — short for **AN**alysis **O**f **VA**riance), is similar to the previous scores, as it measures the dependence of two variables.

ANOVA assumes a linear relationship between the variables and the target, and also that the variables are normally distributed.

It’s well-suited for continuous variables and requires a binary target, but sklearn extends it to regression problems, also.

Here’s a code sample:

The above example applies to classification tasks, but we can use ANOVA for regression tasks via the [f\_regression](http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_regression.html)function provided in the sklearn library.

## **Univariate ROC-AUC /RMSE**

This method uses machine learning models to measure the dependence of two variables. It’s suitable for all variables, and also makes no assumptions about their distribution.

The measure here depends on the problem: we use **RMSE**for regression problems and **ROC-AUC** for classification problems.

The procedure is as follows:

* Build a [decision tree](https://heartbeat.fritz.ai/introduction-to-decision-tree-learning-cd604f85e236) using a single variable and target.
* Rank features according to the model RMSE or ROC-AUC
* Select the features with higher ranking scores.

Here is a code sample for classification problem:

You can also use this method for regression problems, but this time you should change the metrics to something more suitable like RMSE, and of course the algorithm to a regressor.