# Feature Selection

Feature selection is also called variable selection or attribute selection. It is the automatic selection of attributes in data (such as columns in tabular data) that are most relevant to the predictive modeling problem you are working on.

Feature selection is different from dimensionality reduction / Feature extraction. Both methods seek to reduce the number of attributes in the dataset, but a Feature extraction method do so by creating new combinations of attributes, whereas feature selection methods include and exclude attributes present in the data without changing them.

Examples of dimensionality reduction methods include Principal Component Analysis, Singular Value Decomposition and Sammon’s Mapping.

### ****Benefits of performing feature selection before modeling data****

* **Reduces Overfitting**: Less redundant data means less opportunity to make decisions based on noise.
* **Improves Accuracy**: Less misleading data means modeling accuracy improves.
* **Reduces Training Time**: fewer data points reduce algorithm complexity and algorithms train faster.

Feature selection involves two steps, one is searching strategy which select best features from the feature pool and last one is evaluate strategy will actually predict feature importance.

Search strategies contains methods like Optimum, Heuristic & Redundant. Backward & Forward elimination criteria are part of heuristics method. Evaluation strategy contains mainly Supervised / Wrapper methods & unsupervised / filter criteria plus Embedded methods also available.

There are three general classes of feature selection algorithms: filter methods, wrapper methods and embedded methods.

### Filter Methods

Filter feature selection methods apply a statistical measure to assign a scoring to each feature. The features are ranked by the score and either selected to be kept or removed from the dataset. The methods are often univariate and consider the feature independently, or with regard to the dependent variable.

Some examples of some filter methods include the Chi squared test, information gain and correlation coefficient scores.

As the name suggest, in this method, you filter and take only the subset of the relevant features. The model is built after selecting the features. The filtering here is done using correlation matrix and it is most commonly done using Pearson correlation. As the name suggest, in this method, you filter and take only the subset of the relevant features. The model is built after selecting the features. The filtering here is done using correlation matrix and it is most commonly done using Pearson correlation.

cor = df.corr()

#Correlation with output variable  
cor\_target = abs(cor["MEDV"])#Selecting highly correlated features  
relevant\_features = cor\_target[cor\_target>0.5]  
relevant\_features

### Wrapper Methods

Wrapper methods consider the selection of a set of features as a search problem, where different combinations are prepared, evaluated and compared to other combinations. A predictive model us used to evaluate a combination of features and assign a score based on model accuracy.

The search process may be methodical such as a best-first search, it may stochastic such as a random hill-climbing algorithm, or it may use heuristics, like forward and backward passes to add and remove features.

An example if a wrapper method is the recursive feature elimination algorithm.

A wrapper method needs one machine learning algorithm and uses its performance as evaluation criteria. This means, you feed the features to the selected Machine Learning algorithm and based on the model performance you add/remove the features. This is an iterative and computationally expensive process but it is more accurate than the filter method.

There are different wrapper methods such as Backward Elimination, Forward Selection, Bidirectional Elimination and RFE. We will discuss Backward Elimination and RFE here.

* **Backward Elimination**

As the name suggest, we feed all the possible features to the model at first. We check the performance of the model and then iteratively remove the worst performing features one by one till the overall performance of the model comes in acceptable range.

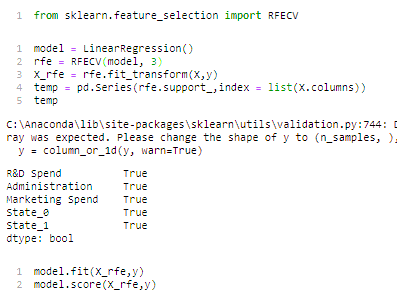
The performance metric used here to evaluate feature performance is pvalue. If the pvalue is above 0.05 then we remove the feature, else we keep it.

We will first run one iteration here just to get an idea of the concept and then we will run the same code in a loop, which will give the final set of features. Here we are using OLS model which stands for “Ordinary Least Squares”. This model is used for performing linear regression.

### 

* **RFE (Recursive Feature Elimination)**

The Recursive Feature Elimination (RFE) method works by recursively removing attributes and building a model on those attributes that remain. It uses accuracy metric to rank the feature according to their importance. The RFE method takes the model to be used and the number of required features as input. It then gives the ranking of all the variables, 1 being most important. It also gives its support, True being relevant feature and False being irrelevant feature.



### Embedded Methods

Embedded methods learn which features best contribute to the accuracy of the model while the model is being created. The most common type of embedded feature selection methods are regularization methods.

Regularization methods are also called penalization methods that introduce additional constraints into the optimization of a predictive algorithm (such as a regression algorithm) that bias the model toward lower complexity (fewer coefficients).

Examples of regularization algorithms are the LASSO, Elastic Net and Ridge Regression.

It is common to use correlation type statistical measures between input and output variables as the basis for filter feature selection. As such, the choice of statistical measures is highly dependent upon the variable data types.

Common data types include numerical (such as height) and categorical (such as a label), although each may be further subdivided such as integer and floating point for numerical variables, and Boolean, ordinal, or nominal for categorical variables.

Embedded methods are iterative in a sense that takes care of each iteration of the model training process and carefully extract those features which contribute the most to the training for a particular iteration. Regularization methods are the most commonly used embedded methods which penalize a feature given a coefficient threshold.

Here we will do feature selection using Lasso regularization. If the feature is irrelevant, lasso penalizes its coefficient and make it 0. Hence the features with coefficient = 0 are removed and the rest are taken.

### Common input variable data types:

* **Numerical Variables**
* Integer Variables.
* Floating Point Variables.
* **Categorical Variables**.
* Boolean Variables (dichotomous).
* Ordinal Variables.
* Nominal Variables.

The more that is known about the data type of a variable, the easier it is to choose an appropriate statistical measure for a filter-based feature selection method.

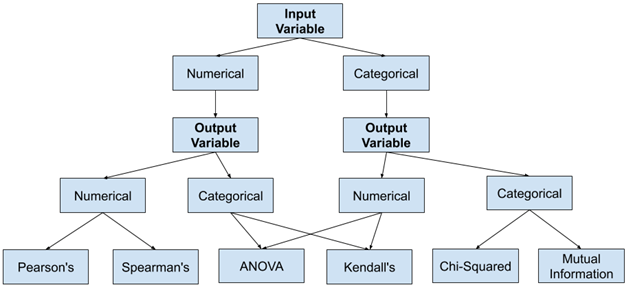
### Statistics for Filter-Based Feature Selection Methods

Input variables are those that are provided as input to a model. In feature selection, it is this group of variables that we wish to reduce in size. Output variables are those for which a model is intended to predict, often called the response variable.

The type of response variable typically indicates the type of predictive modeling problem being performed. For example, a numerical output variable indicates a regression predictive modeling problem, and a categorical output variable indicates a classification predictive modeling problem.

1. **Numerical Output**: Regression predictive modeling problem.
2. **Categorical Output**: Classification predictive modeling problem.

The statistical measures used in filter-based feature selection are generally calculated one input variable at a time with the target variable. As such, they are referred to as univariate statistical measures. This may mean that any interaction between input variables is not considered in the filtering process.



### Numerical Input, Numerical Output

This is a regression predictive modeling problem with numerical input variables.

The most common techniques are to use a correlation coefficient, such as Pearson’s for a linear correlation, or rank-based methods for a nonlinear correlation.

* Pearson’s correlation coefficient (linear).
* Spearman’s rank coefficient (nonlinear)

### Numerical Input, Categorical Output

This is a classification predictive modeling problem with numerical input variables.

This might be the most common example of a classification problem,

Again, the most common techniques are correlation based, although in this case, they must take the categorical target into account.

* ANOVA correlation coefficient (linear).
* Kendall’s rank coefficient (nonlinear).

Kendall does assume that the categorical variable is ordinal.

### Categorical Input, Numerical Output

This is a regression predictive modeling problem with categorical input variables. This is a strange example of a regression problem (e.g. you would not encounter it often).

Nevertheless, you can use the same “Numerical Input, Categorical Output” methods (described above), but in reverse.

### Categorical Input, Categorical Output

This is a classification predictive modeling problem with categorical input variables.

The most common correlation measure for categorical data is the chi-squared test.  You can also use mutual information (information gain) from the field of information theory.

* Chi-Squared test (contingency tables).
* Mutual Information.

In fact, mutual information is a powerful method that may prove useful for both categorical and numerical data, e.g. it is agnostic to the data types.

### Correlation Statistics

The scikit-learn library provides an implementation of most of the useful statistical measures.

For example:

* Pearson’s Correlation Coefficient: [f\_regression()](https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_regression.html)
* ANOVA: [f\_classif()](https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_classif.html)
* Chi-Squared: [chi2()](https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.chi2.html)

Mutual Information: [mutual\_info\_classif()](https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_classif.html) and [mutual\_info\_regression()](https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_regression.html)

Also, the SciPy library provides an implementation of many more statistics, such as Kendall’s tau ([kendalltau](https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kendalltau.html)) and Spearman’s rank correlation ([spearmanr](https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.spearmanr.html)).

### Selection Method

The scikit-learn library also provides many different filtering methods once statistics have been calculated for each input variable with the target.

Two of the more popular methods include:

* Select the top k variables: [SelectKBest](https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html)
* Select the top percentile variables: [SelectPercentile](https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectPercentile.html)