Feature Selection

19 October 2023 21:38

Feature selection is the process of reducing the number of input variables when developing a predictive model.
It is desirable to reduce the number of input variables to both reduce the computational cost of modelling and, in some cases, to improve the performance of the model.

Statistical-based feature selection methods involve evaluating the relationship between each input variable and the target variable using statistics and selecting those input variables that have the strongest relationship with the target variable.

1. Feature Selection Methods

Feature selection methods are intended to reduce the number of input variables to those that are believed to be most useful to a model in order to predict the target variable.

Some predictive modeling problems have a large number of variables that can slow the development and training of models and require a large amount of system memory.
Many models, especially those based on regression slopes and intercepts, will estimate parameters for every term in the model. Because of this, the presence of non-informative variables can add uncertainty to the predictions and reduce the overall effectiveness of the model.

One way to think about feature selection methods are in terms of **supervised** and **unsupervised** methods.

The difference has to do with whether features are selected based on the target variable or not. Unsupervised feature selection techniques ignores the target variable, such as methods that remove redundant variables using correlation. Supervised feature selection techniques use the target variable, such as methods that remove irrelevant variables.

Another way to consider the mechanism used to select features which may be divided into **wrapper** and **filter** methods. These methods are almost always supervised and are evaluated based on the performance of a resulting model on a holdout dataset.

Wrapper feature selection methods create many models with different subsets of input features and select those features that result in the best performing model according to a performance metric. These methods are unconcerned with the variable types, although they can be computationally expensive. RFE is a good example of a wrapper feature selection method.

Wrapper methods evaluate multiple models using procedures that add and/or remove predictors to find the optimal combination that maximizes model performance.

Filter feature selection methods use statistical techniques to evaluate the relationship between each input variable and the target variable, and these scores are used as the basis to choose (filter) those input variables that will be used in the model.

Filter methods evaluate the relevance of the predictors outside of the predictive models and subsequently model only the predictors that pass some criterion.

Finally, there are some machine learning algorithms that perform feature selection automatically as part of learning the model. We might refer to these techniques as **intrinsic** feature selection methods.
… some models contain built-in feature selection, meaning that the model will only include predictors that help maximize accuracy. In these cases, the model can pick and choose which representation of the data is best.

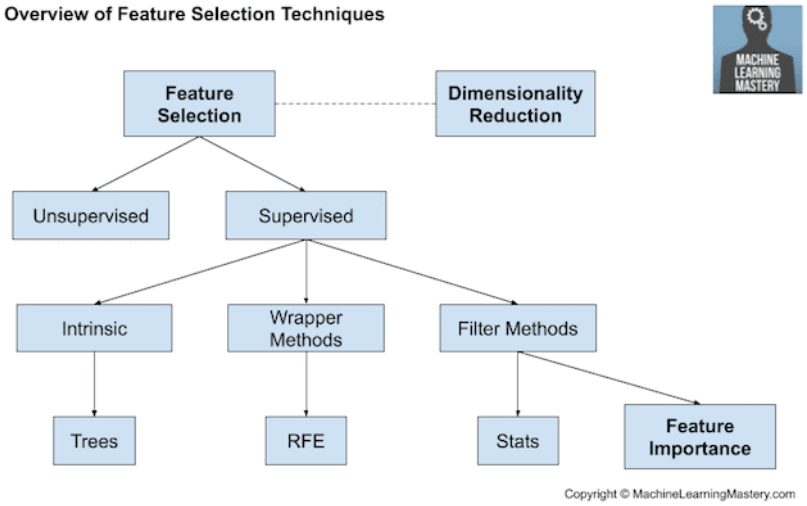
This includes algorithms such as penalized regression models like Lasso and decision trees, including ensembles of decision trees like random forest.
Some models are naturally resistant to non-informative predictors. Tree- and rule-based models, MARS and the lasso, for example, intrinsically conduct feature selection.

Feature selection is also related to dimensionally reduction techniques in that both methods seek fewer input variables to a predictive model. The difference is that feature selection select features to keep or remove from the dataset, whereas dimensionality reduction create a projection of the data resulting in entirely new input features. As such, dimensionality reduction is an alternate to feature selection rather than a type of feature selection.

We can summarize feature selection as follows.

- **Feature Selection**: Select a subset of input features from the dataset.

- **Unsupervised**: Do not use the target variable (e.g. remove redundant variables).
- Correlation
- **Supervised**: Use the target variable (e.g. remove irrelevant variables).
- **Wrapper**: Search for well-performing subsets of features.
- RFE
- **Filter**: Select subsets of features based on their relationship with the target.
- Statistical Methods
- Feature Importance Methods
- **Intrinsic**: Algorithms that perform automatic feature selection during training.
- Decision Trees
- **Dimensionality Reduction**: Project input data into a lower-dimensional feature space.



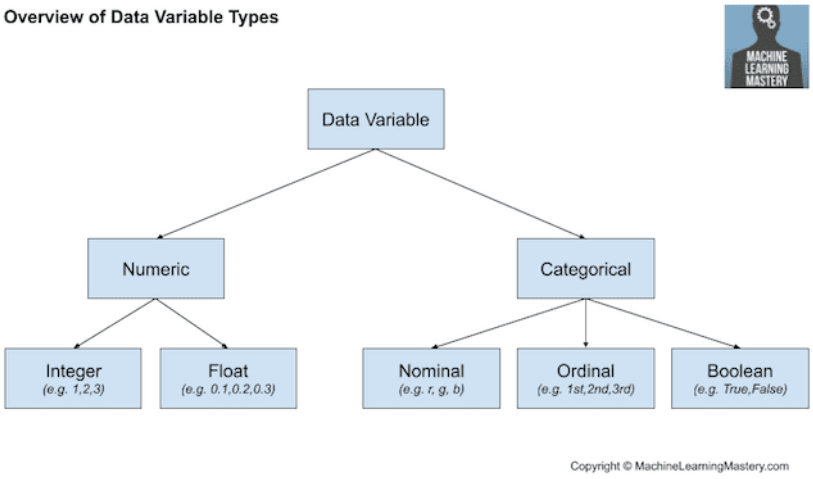
2. Statistics for Filter-Based Feature Selection Methods

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.

Common input variable data types:

- **Numerical Variables**
- Integer Variables.
- Floating Point Variables.

- **Categorical Variables**.
- Boolean Variables (dichotomous).
- Ordinal Variables.
- Nominal Variables.



In this section, we will consider two broad categories of variable types: numerical and categorical; also, the two main groups of variables to consider: input and output.

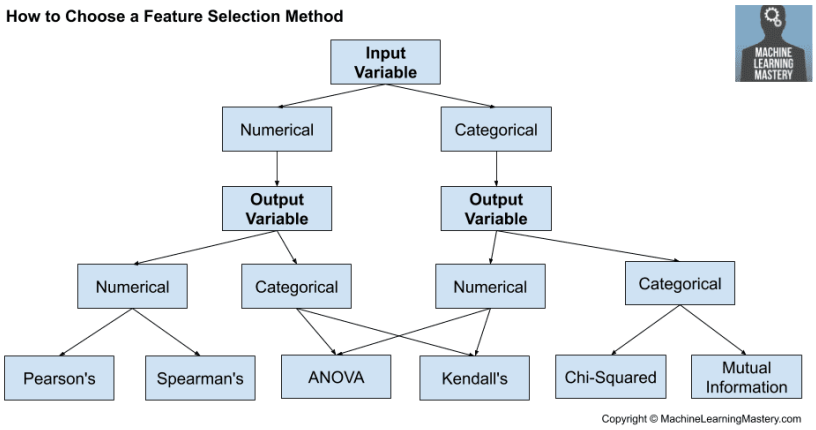
Input variables are those that are provided as input to a model. In feature selection, it is this group of variables that we wish to reduce in size.

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.

- **Numerical Output**: Regression predictive modeling problem.
- **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.
Most of these techniques are univariate, meaning that they evaluate each predictor in isolation. In this case, the existence of correlated predictors makes it possible to select important, but redundant, predictors. The obvious consequences of this issue are that too many predictors are chosen and, as a result, collinearity problems arise.

With this framework, let's review some univariate statistical measures that can be used for filter-based feature selection.



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.

3. Tips and Tricks for Feature Selection

This section provides some additional considerations when using filter-based feature selection.

Correlation Statistics

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

- Pearson's Correlation Coefficient: f_regression()
- ANOVA: f_classif()
- Chi-Squared: chi2()
- Mutual Information: mutual_info_classif() and mutual_info_regression()

Also, the SciPy library provides an implementation of many more statistics, such as Kendall's tau (kendalltau) and Spearman's rank correlation (spearmanr).

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
- Select the top percentile variables: SelectPercentile

I often use *SelectKBest* myself.

Transform Variables

Consider transforming the variables in order to access different statistical methods.
For example, you can transform a categorical variable to ordinal, even if it is not, and see if any interesting results come out.
You can also make a numerical variable discrete (e.g. bins); try categorical-based measures.
Some statistical measures assume properties of the variables, such as Pearson's that assumes a Gaussian probability distribution to the observations and a linear relationship. You can transform the data to meet the expectations of the test and try the test regardless of the expectations and compare results.

What Is the Best Method?

There is no best feature selection method.
Just like there is no best set of input variables or best machine learning algorithm. At least not universally.
Instead, you must discover what works best for your specific problem using careful systematic experimentation.
Try a range of different models fit on different subsets of features chosen via different statistical measures and discover what works best for your specific problem.

4. Worked Examples of Feature Selection

It can be helpful to have some worked examples that you can copy-and-paste and adapt for your own project.
This section provides worked examples of feature selection cases that you can use as a starting point.

Regression Feature Selection: (*Numerical Input, Numerical Output*)

This section demonstrates feature selection for a regression problem that as numerical inputs and numerical outputs.
A test regression problem is prepared using the make_regression() function.
Feature selection is performed using Pearson's Correlation Coefficient via the f_regression() function.

```
# pearson's correlation feature selection for numeric input and numeric output
from sklearn.datasets import make_regression
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_regression
# generate dataset
X, y = make_regression(n_samples=100, n_features=100, n_informative=10)
# define feature selection
fs = SelectKBest(score_func=f_regression, k=10)
# apply feature selection
X_selected = fs.fit_transform(X, y)
print(X_selected.shape)
```

Running the example first creates the regression dataset, then defines the feature selection and applies the feature selection procedure to the dataset, returning a subset of the selected input features.

```
(100, 10)
```

Classification Feature Selection:
(*Numerical Input, Categorical Output*)

This section demonstrates feature selection for a classification problem that as numerical inputs and categorical outputs.
A test regression problem is prepared using the make_classification() function.
Feature selection is performed using ANOVA F measure via the f_classif() function.

```
# ANOVA feature selection for numeric input and categorical output
from sklearn.datasets import make_classification
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_classif
# generate dataset
X, y = make_classification(n_samples=100, n_features=20, n_informative=2)
# define feature selection
fs = SelectKBest(score_func=f_classif, k=2)
# apply feature selection
X_selected = fs.fit_transform(X, y)
print(X_selected.shape)
```

Running the example first creates the classification dataset, then defines the feature selection and applies the feature selection procedure to the dataset, returning a subset of the selected input features.

```
(100, 2)
```