**An Introduction to Feature Selection**

by [**Jason Brownlee**](https://machinelearningmastery.com/author/jasonb/) on October 6, 2014 in [**Machine Learning Process**](https://machinelearningmastery.com/category/machine-learning-process/)

Tweet  **Share** **Share**

Last Updated on June 3, 2019

Which features should you use to create a [predictive model](https://machinelearningmastery.com/gentle-introduction-to-predictive-modeling/)?

This is a difficult question that may require deep knowledge of the problem domain.

It is possible to automatically select those features in your data that are most useful or most relevant for the problem you are working on. This is a process called feature selection.

In this post you will discover feature selection, the types of methods that you can use and a handy checklist that you can follow the next time that you need to select features for a machine learning model.

[](https://3qeqpr26caki16dnhd19sv6by6v-wpengine.netdna-ssl.com/wp-content/uploads/2014/10/feature-selection.jpg)

An Introduction to Feature Selection  
Photo by [John Tann](https://www.flickr.com/photos/31031835@N08/6498604953), some rights reserved

**What is Feature Selection**

Feature selection is also called variable selection or attribute selection.

It is the automatic selection of attributes in your data (such as columns in tabular data) that are most relevant to the predictive modeling problem you are working on.

*feature selection… is the process of selecting a subset of relevant features for use in model construction*

— [Feature Selection](http://en.wikipedia.org/wiki/Feature_selection), Wikipedia entry.

Feature selection is different from dimensionality reduction. Both methods seek to reduce the number of attributes in the dataset, but a dimensionality reduction method do so by creating new combinations of attributes, where as 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.

*Feature selection is itself useful, but it mostly acts as a filter, muting out features that aren’t useful in addition to your existing features.*

— Robert Neuhaus, in answer to “[How valuable do you think feature selection is in machine learning?](http://www.quora.com/How-valuable-do-you-think-feature-selection-is-in-machine-learning-Which-do-you-think-improves-accuracy-more-feature-selection-or-feature-engineering)”

**The Problem The Feature Selection Solves**

Feature selection methods aid you in your mission to create an accurate predictive model. They help you by choosing features that will give you as good or better accuracy whilst requiring less data.

Feature selection methods can be used to identify and remove unneeded, irrelevant and redundant attributes from data that do not contribute to the accuracy of a predictive model or may in fact decrease the accuracy of the model.

Fewer attributes is desirable because it reduces the complexity of the model, and a simpler model is simpler to understand and explain.

*The objective of variable selection is three-fold: improving the prediction performance of the predictors, providing faster and more cost-effective predictors, and providing a better understanding of the underlying process that generated the data.*

— Guyon and Elisseeff in “[An Introduction to Variable and Feature Selection](http://jmlr.csail.mit.edu/papers/volume3/guyon03a/guyon03a.pdf)” (PDF)

**Feature Selection Algorithms**

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](https://machinelearningmastery.com/how-to-use-correlation-to-understand-the-relationship-between-variables/).

**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.

**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.

**Feature Selection Tutorials and Recipes**

We have seen a number of examples of features selection before on this blog.

* **Weka**: For a tutorial showing how to perform feature selection using Weka see “[Feature Selection to Improve Accuracy and Decrease Training Time](http://machinelearningmastery.com/feature-selection-to-improve-accuracy-and-decrease-training-time/)“.
* **Scikit-Learn**: For a recipe of Recursive Feature Elimination in Python using scikit-learn, see “[Feature Selection in Python with Scikit-Learn](http://machinelearningmastery.com/feature-selection-in-python-with-scikit-learn/)“.
* **R**: For a recipe of Recursive Feature Elimination using the Caret R package, see “[Feature Selection with the Caret R Package](http://machinelearningmastery.com/feature-selection-with-the-caret-r-package/)“

**A Trap When Selecting Features**

Feature selection is another key part of the applied machine learning process, like model selection. You cannot fire and forget.

It is important to consider feature selection a part of the model selection process. If you do not, you may inadvertently introduce bias into your models which can result in overfitting.

*… should do feature selection on a different dataset than you train [your predictive model] on … the effect of not doing this is you will overfit your training data.*

— Ben Allison in answer to “[Is using the same data for feature selection and cross-validation biased or not?](http://stats.stackexchange.com/questions/40576/is-using-the-same-data-for-feature-selection-and-cross-validation-biased-or-not)”

For example, you must include feature selection within the inner-loop when you are using accuracy estimation methods such as cross-validation. This means that feature selection is performed on the prepared fold right before the model is trained. A mistake would be to perform feature selection first to prepare your data, then perform model selection and training on the selected features.

*If we adopt the proper procedure, and perform feature selection in each fold, there is no longer any information about the held out cases in the choice of features used in that fold.*

— Dikran Marsupial in answer to “[Feature selection for final model when performing cross-validation in machine learning](http://stats.stackexchange.com/questions/2306/feature-selection-for-final-model-when-performing-cross-validation-in-machine)”

The reason is that the decisions made to select the features were made on the entire training set, that in turn are passed onto the model. This may cause a mode a model that is enhanced by the selected features over other models being tested to get seemingly better results, when in fact it is biased result.

*If you perform feature selection on all of the data and then cross-validate, then the test data in each fold of the cross-validation procedure was also used to choose the features and this is what biases the performance analysis.*

— Dikran Marsupial in answer to “[Feature selection and cross-validation](http://stats.stackexchange.com/questions/27750/feature-selection-and-cross-validation)”

**Feature Selection Checklist**

Isabelle Guyon and Andre Elisseeff the authors of “[An Introduction to Variable and Feature Selection](http://jmlr.csail.mit.edu/papers/volume3/guyon03a/guyon03a.pdf)” (PDF) provide an excellent checklist that you can use the next time you need to select data features for you predictive modeling problem.

I have reproduced the salient parts of the checklist here:

1. **Do you have domain knowledge?** If yes, construct a better set of ad hoc”” features
2. **Are your features commensurate?** If no, consider normalizing them.
3. **Do you suspect interdependence of features?** If yes, expand your feature set by constructing conjunctive features or products of features, as much as your computer resources allow you.
4. **Do you need to prune the input variables (e.g. for cost, speed or data understanding reasons)?** If no, construct disjunctive features or weighted sums of feature
5. **Do you need to assess features individually (e.g. to understand their influence on the system or because their number is so large that you need to do a first filtering)?** If yes, use a variable ranking method; else, do it anyway to get baseline results.
6. **Do you need a predictor?** If no, stop
7. **Do you suspect your data is “dirty” (has a few meaningless input patterns and/or noisy outputs or wrong class labels)?** If yes, detect the outlier examples using the top ranking variables obtained in step 5 as representation; check and/or discard them.
8. **Do you know what to try first?** If no, use a linear predictor. Use a forward selection method with the “probe” method as a stopping criterion or use the 0-norm embedded method for comparison, following the ranking of step 5, construct a sequence of predictors of same nature using increasing subsets of features. Can you match or improve performance with a smaller subset? If yes, try a non-linear predictor with that subset.
9. **Do you have new ideas, time, computational resources, and enough examples?** If yes, compare several feature selection methods, including your new idea, correlation coefficients, backward selection and embedded methods. Use linear and non-linear predictors. Select the best approach with model selection
10. **Do you want a stable solution (to improve performance and/or understanding)?** If yes, subsample your data and redo your analysis for several “bootstrap”.

**Further Reading**

Do you need help with feature selection on a specific platform? Below are some tutorials that can get you started fast:

* [How to perform feature selection in Weka](http://machinelearningmastery.com/perform-feature-selection-machine-learning-data-weka/) (without code)
* [How to perform feature selection in Python with scikit-learn](http://machinelearningmastery.com/feature-selection-machine-learning-python/)
* [How to perform feature selection in R with caret](http://machinelearningmastery.com/feature-selection-with-the-caret-r-package/)

To go deeper into the topic, you could pick up a dedicated book on the topic, such as any of the following:

* [Feature Selection for Knowledge Discovery and Data Mining](http://www.amazon.com/dp/079238198X?tag=inspiredalgor-20)
* [Computational Methods of Feature Selection](http://www.amazon.com/dp/1584888784?tag=inspiredalgor-20)
* [Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches](http://www.amazon.com/dp/0470229756?tag=inspiredalgor-20)
* [Subspace, Latent Structure and Feature Selection: Statistical and Optimization Perspectives Workshop](http://www.amazon.com/dp/3540341374?tag=inspiredalgor-20)
* [Feature Extraction, Construction and Selection: A Data Mining Perspective](http://www.amazon.com/dp/0792381963?tag=inspiredalgor-20)

Feature Selection is a sub-topic of Feature Engineering. You might like to take a deeper look at feature engineering in the post: ”