<https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/>

Attribute selection: The goal of feature selection is two-fold - We want to improve the computational efficiency and reduce the generalization error of the model by removing irrelevant features or noise.

**Filter methods** are generally used as a preprocessing step. The selection of attributes/features is independent of any machine learning algorithms. Instead, features are selected on the basis of their scores in various statistical tests for their correlation with the outcome variable.

In **wrapper methods**, we try to use a subset of features and train a model using them. Based on the inferences that we draw from the previous model, we decide to add or remove features from your subset. The problem is essentially reduced to a search problem.

**Embedded methods** combine the best of the two above. Lasso L1 regularization is an example of this. Wrapper and Embedded methods give more accurate results but as they are computationally expensive, a new model is built and tested for each attribute. These methods are best suited when you have not many (~20) attributes.

The main differences between the filter and wrapper methods for feature selection are:

* Filter methods measure the relevance of features by their correlation with dependent variable while wrapper methods measure the usefulness of a subset of feature by actually training a model on it.
* Filter methods are much faster compared to wrapper methods as they do not involve training the models. On the other hand, wrapper methods are computationally very expensive as well.
* Filter methods use statistical methods for evaluation of a subset of features while wrapper methods use cross validation.
* Filter methods might fail to find the best subset of features in many occasions, but wrapper methods can always provide the best subset of features.
* Using the subset of features from the wrapper methods make the model more prone to overfitting as compared to using subset of features from the filter methods.

Table 1: Attribute Filtering using sklearn feature\_selection, SelectPercentile

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Top correlated attributes to the class label used in model training and prediction** | Validation Set + Test Set Accuracy / 2, over 5 trials, using DNNClassifier hidden units [37, 30, 19], 1000 steps, and Adam optimizer learning rate 0.001 | | | | | **Average Accuracy** |
| **10% (137 attributes)** | 0.61 | 0.70 | 0.66 | 0.78 | 0.66 | **0.68** |
| **20% (274 attributes)** | 0.73 | 0.77 | 0.64 | 0.78 | 0.76 | **0.74** |
| **30% (410 attributes)** | 0.81 | 0.84 | 0.82 | 0.82 | 0.88 | **0.83** |
| **40% (547 attributes)** | 0.89 | 0.82 | 0.84 | 0.80 | 0.84 | **0.84** |
| **50% (683 attributes)** | 0.80 | 0.89 | 0.89 | 0.80 | 0.80 | **0.84** |
| **60% (820 attributes)** | 0.88 | 0.92 | 0.89 | 0.85 | 0.88 | **0.88** |
| **70% (957 attributes)** | 0.95 | 0.89 | 0.87 | 0.95 | 0.85 | **0.90** |
| **80% (1093 attributes)** | 0.92 | 0.92 | 0.97 | 0.86 | 0.88 | **0.91** |
| **90% (1230 attributes)** | 0.95 | 0.87 | 0.86 | 0.87 | 0.86 | **0.88** |
| **100% (all 1366 attributes)** | 0.94 | 0.86 | 0.86 | 0.94 | 0.87 | **0.89** |

Table 1: T-Test Filtering using scipy stats.ttest\_ind, determining the most correlated features to the class label.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Top correlated attributes to the class label used in model training and prediction** | Averaged validation & test set accuracy over 5 trials, using sklearn.svm with a linear kernel. | | | | | **Average Accuracy** |
| **40% (547 attributes)** | 0.56 | 0.49 | 0.55 | 0.59 | 0.63 | **0.564** |
| **50% (683 attributes)** | 0.58 | 0.81 | 0.83 | 0.67 | 0.63 | **0.704** |
| **60% (820 attributes)** | 0.72 | 0.64 | 0.77 | 0.71 | 0.62 | **0.692** |
| **70% (957 attributes)** | 0.73 | 0.78 | 0.68 | 0.79 | 0.78 | **0.752** |
| **80% (1093 attributes)** | 0.82 | 0.85 | 0.79 | 0.83 | 0.69 | **0.796** |
| **90% (1230 attributes)** | 0.88 | 0.90 | 0.88 | 0.85 | 0.94 | **0.890** |
| **100% (all 1366 attributes)** | 0.90 | 0.87 | 0.89 | 0.94 | 0.87 | **0.894** |

x is the data in column “45”, y is the distribution.



