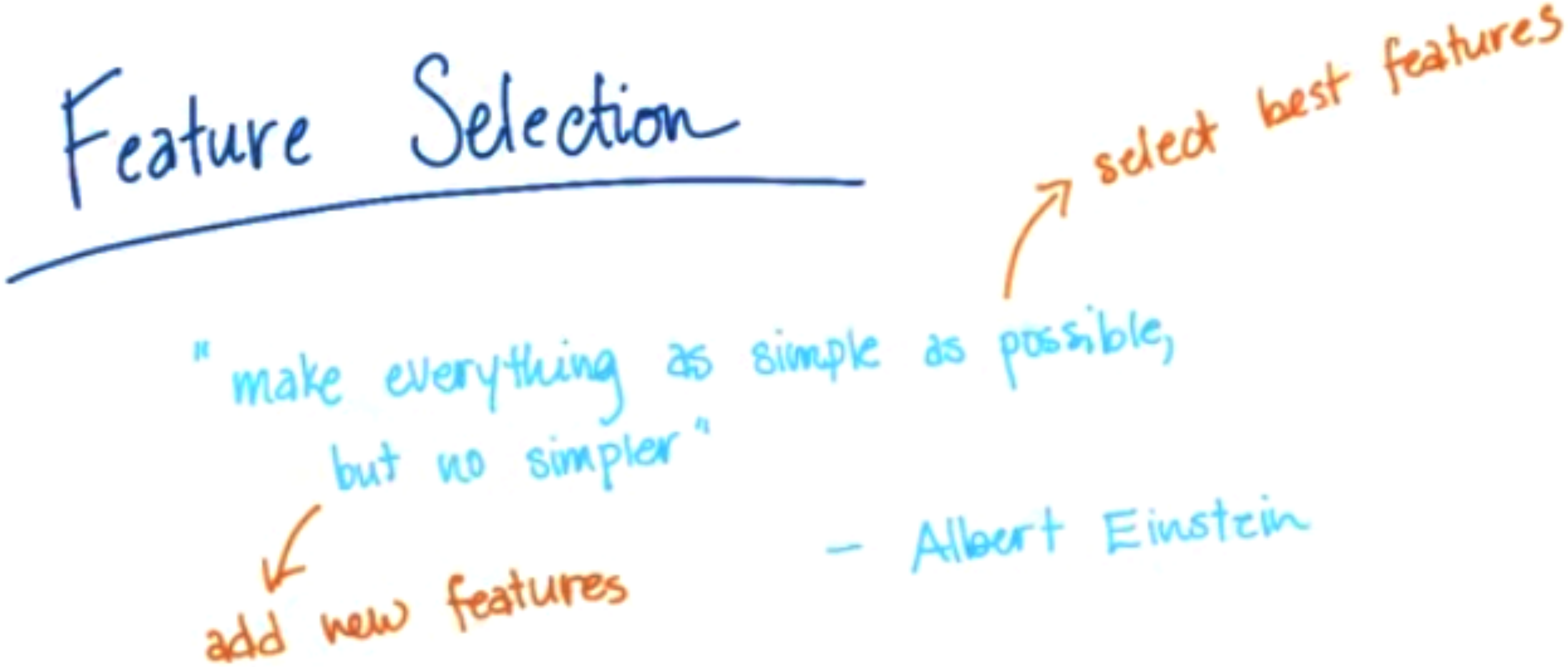
Feature Selection

- Feature selection is the process of selecting a subset of relevant features (variables, predictors) for use in model construction.

- Also known as
    - variable selection
    - attribute selection
    - variable subset selection



Feature Selection

"make everything as simple as possible, but no simpler"

— Albert Einstein

- https://en.wikipedia.org/wiki/Feature_selection

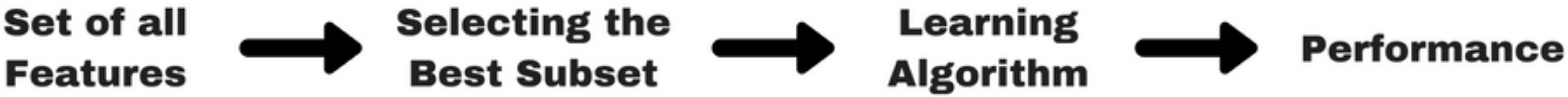
Importance of Feature Selection

- It enables the machine learning algorithm to train faster.
- It reduces the complexity of a model and makes it easier to interpret.
- It improves the accuracy of a model if the right subset is chosen.
- It reduces overfitting.

Feature Selection methods

- Filter methods
- Wrapper methods

Filter methods



Set of all Features → Selecting the Best Subset → Learning Algorithm → Performance

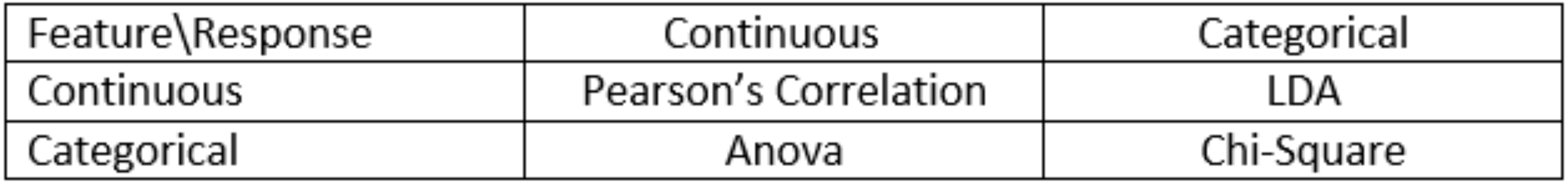
- Filter methods are generally used as a preprocessing step.

- The selection of 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.

  - The correlation is a subjective term here.

- Filter methods do not remove multicollinearity. So, you must deal with multicollinearity of features as well before training models for your data.



Feature\Response	Continuous	Categorical
Continuous	Pearson’s Correlation	LDA
Categorical	Anova	Chi-Square

Filter methods

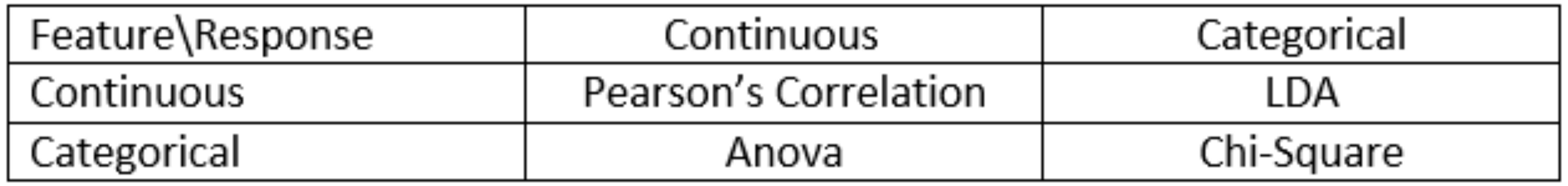
- **Pearson’s Correlation**: It is used as a measure for quantifying linear dependence between two continuous variables X and Y. Its value varies from -1 to +1. Pearson’s correlation is given as:

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y}$$

- **LDA**: Linear discriminant analysis is used to find a linear combination of features that characterizes or separates two or more classes (or levels) of a categorical variable.

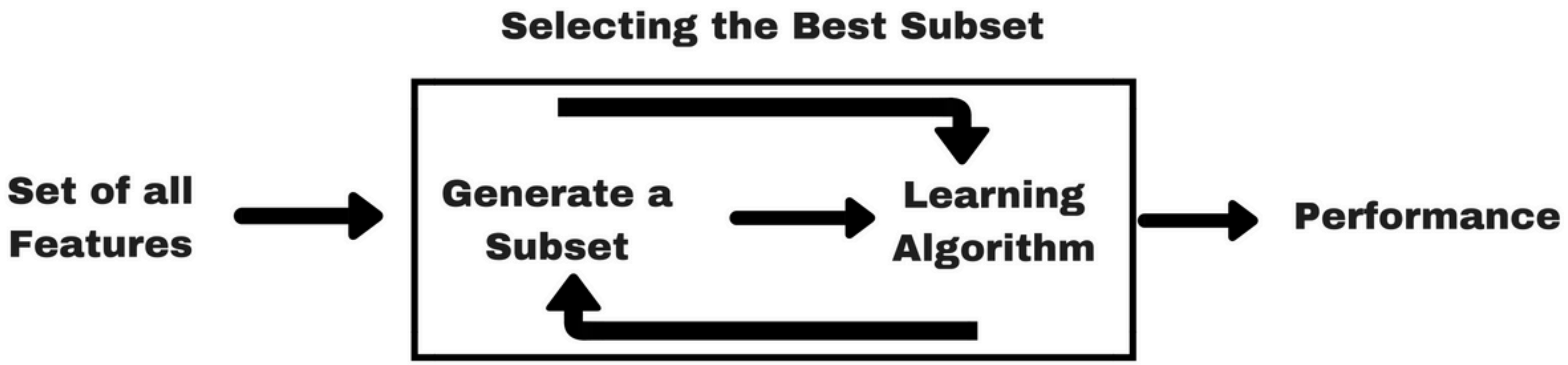
- **ANOVA**: ANOVA stands for Analysis of variance. It is similar to LDA except for the fact that it is operated using one or more categorical independent features and one continuous dependent feature. It provides a statistical test of whether the means of several groups are equal or not.

- **Chi-Square**: It is a is a statistical test applied to the groups of categorical features to evaluate the likelihood of correlation or association between them using their frequency distribution.



Feature\Response	Continuous	Categorical
Continuous	Pearson’s Correlation	LDA
Categorical	Anova	Chi-Square

Wrapper methods



- In wrapper methods, a subset of features is used to train a model.

- 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.

- Wrapper methods are usually computationally very expensive.

Examples of Wrapper methods

- **Forward Selection**
  - Forward selection is an iterative method in which we start with having no feature in the model. In each iteration, we keep adding the feature which best improves our model till an addition of a new variable does not improve the performance of the model.
- **Backward Elimination**
  - In backward elimination, we start with all the features and removes the least significant feature at each iteration which improves the performance of the model. We repeat this until no improvement is observed on removal of features.
- **Recursive Feature elimination**
  - It is a greedy optimisation algorithm which aims to find the best performing feature subset. It repeatedly creates models and keeps aside the best or the worst performing feature at each iteration. It constructs the next model with the left features until all the features are exhausted. It then ranks the features based on the order of their elimination.

Recursive Feature Elimination

```
# Feature Extraction with RFE

from pandas import read_csv

from sklearn.feature_selection import RFE

from sklearn.linear_model import LogisticRegression
```

Recursive Feature Elimination

```
# load data

url = "https://archive.ics.uci.edu/ml/machine-learning-databases/
pima-indians-diabetes/pima-indians-diabetes.data"

names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi',
'age', 'class']

dataframe = read_csv(url, names=names)

array = dataframe.values

X = array[:,0:8]

Y = array[:,8]
```

Recursive Feature Elimination

```
# feature extraction

model = LogisticRegression()

rfe = RFE(model, 3)

fit = rfe.fit(X, Y)

print("Num Features: %d") % fit.n_features_

print("Selected Features: %s") % fit.support_

print("Feature Ranking: %s") % fit.ranking_
```

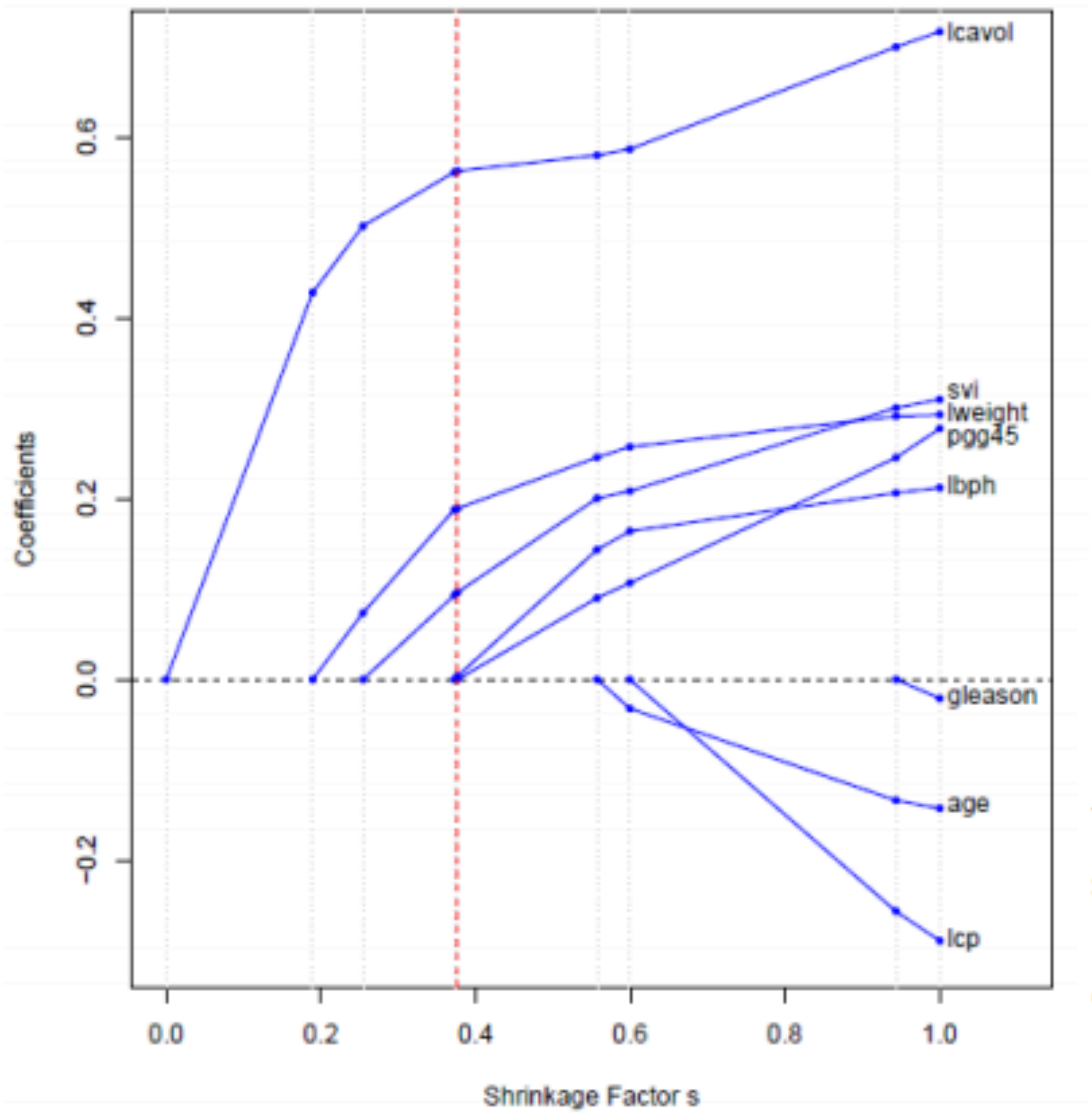
Feature elimination using the lasso penalty

- LASSO minimizes the Residual Sum of Squares (RSS) but poses a constraint to the sum of the absolute values of the coefficients being less than a constant.

- The L1 penalty is a popular method for feature selection. As the regularization strength increases more features will be removed.

- The λ parameter can be tuned in order to set the shrinkage level, the higher the λ is, the more coefficients are shrunk to 0.

Feature elimination using the LASSO penalty



$$s = \frac{\sum|\boldsymbol{\beta}|}{\sum|\hat{\boldsymbol{\beta}}_{LS}|}$$

where $\hat{\boldsymbol{\beta}}_{LS}$ are the OLS coefficients. When $s = 1$, the shrinkage level is zero and the LASSO solution corresponds to the OLS solution; when $s < 1$, LASSO shrinks the coefficients (the lower the s value, the higher the λ value). For certain values of s, some coefficients are shrunk exactly to zero. This path is depicted in Figure 6[2].

Figure 6. Shrinkage of coefficients by LASSO. When s = 1, LASSO and OLS solutions coincide (right side of the graph). When s < 1 regression coefficients are shrunk. The lower the s value is, the more the coefficients are shrunk to 0. For example, for s close to 0.4 (dotted red line), only 3 variables are selected.

http://www.moleculardescriptors.eu/tutorials/T6_moleculardescriptors_variable_selection.pdf

Differences between Filter and Wrapper methods

- 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.

Sci-Kit Learn References

- http://scikit-learn.org/stable/modules/feature_selection.html
- http://scikit-learn.org/stable/auto_examples/feature_selection/plot_feature_selection.html
- http://scikit-learn.org/stable/auto_examples/feature_selection/plot_select_from_model_boston.html
- http://scikit-learn.org/stable/modules/classes.html#module-sklearn.feature_selection

Packages

- https://github.com/scikit-learn-contrib/boruta_py
- https://github.com/jundongl/scikit-feature

More references

- http://blog.datadive.net/selecting-good-features-part-i-univariate-selection
- http://blog.datadive.net/selecting-good-features-part-ii-linear-models-and-regularization/
- http://blog.datadive.net/selecting-good-features-part-iii-random-forests/
- http://blog.datadive.net/selecting-good-features-part-iv-stability-selection-rfe-and-everything-side-by-side/
- https://machinelearningmastery.com/feature-selection-in-python-with-scikit-learn/
- https://machinelearningmastery.com/feature-selection-machine-learning-python/