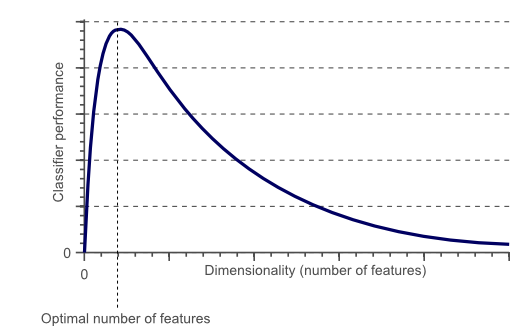
**Dimensionality Reduction**



As the number of features increases, the model becomes more complex. The more the number of features, the more the chances of overfitting. A machine learning model that is trained on many features, gets increasingly dependent on the data it was trained on and in turn overfitted, resulting in poor performance on real data, beating the purpose.

dimensionality reduction advantages

1. Less misleading data means model accuracy improves.
2. Less dimensions mean less computing. Less data means that algorithms train faster.
3. Less data means less storage space required.
4. Less dimensions allow usage of algorithms unfit for a large number of dimensions
5. Removes redundant features and noise.

**Feature Selection and Feature Engineering for dimensionality reduction**

Dimensionality reduction could be done by both feature selection methods as well as feature engineering methods.

Feature selection is the process of identifying and selecting relevant features for your sample.

Feature engineering is manually generating new features from existing features, by applying some transformation or performing some operation on them.

There are a lot of tools could employ to aid feature selection.

1. Heatmaps that show the correlation between features is a good idea.
2. Just visualizing the relationship between the features and the target variable by plotting each feature against the target variable.

Few programmatic methods for feature selection from the popular machine learning library sci-kit learn, namely,

1. Variance Threshold and
2. Univariate selection.

Variance Threshold is a baseline approach to feature selection. As the name suggests, it drops all features where the variance along the column does not exceed a threshold value. The premise is that a feature which doesn’t vary much within itself, has very little predictive power.

Univariate Feature Selection uses statistical tests to select features. Univariate describes a type of data which consists of observations on only a single characteristic or attribute. Univariate feature selection examines each feature individually to determine the strength of the relationship of the feature with the response variable. Some examples of statistical tests that can be used to evaluate feature relevance are Pearson Correlation, Maximal information coefficient, Distance correlation, ANOVA and Chi-square. Chi-square is used to find the relationship between categorical variables and Anova is preferred when the variables are continuous.

**Scikit-learn exposes feature selection routines likes SelectKBest, SelectPercentile or GenericUnivariateSelect as objects that implement a transform method based on the score of anova or chi2 or mutual information.**

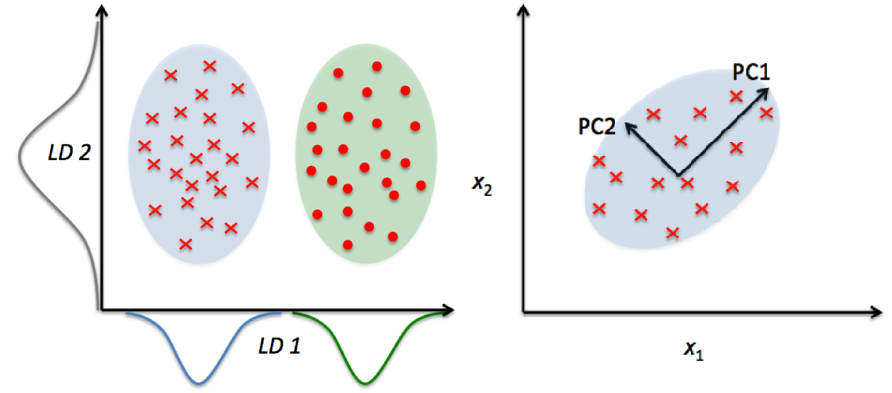
**Sklearn offers f\_regression and mutual\_info\_regression as the scoring functions for regression and f\_classif and mutual\_info\_classif for classification.**

F-Test checks for and only captures linear relationships between features and labels. A highly correlated feature is given higher score and less correlated features are given lower score. Correlation is highly deceptive as it doesn’t capture strong non-linear relationships. On the other hand, mutual information methods can capture any kind of statistical dependency, but being nonparametric, they require more samples for accurate estimation.

#### ****Linear Dimensionality Reduction Methods****

The most common and well-known dimensionality reduction methods are the ones that apply linear transformations, like

1. PCA (Principal Component Analysis) : Popularly used for dimensionality reduction in continuous data, PCA rotates and projects data along the direction of increasing variance. The features with the maximum variance are the principal components.
2. Factor Analysis : a technique that is used to reduce a large number of variables into fewer numbers of factors. The values of observed data are expressed as functions of a number of possible causes in order to find which are the most important. The observations are assumed to be caused by a linear transformation of lower dimensional latent factors and added Gaussian noise.
3. LDA (Linear Discriminant Analysis): projects data in a way that the class separability is maximised. Examples from same class are put closely together by the projection. Examples from different classes are placed far apart by the projection



#### Non-linear Dimensionality Reduction Methods

Non-linear transformation methods or manifold learning methods are used when the data doesn’t lie on a linear subspace. It is based on the manifold hypothesis which says that in a high dimensional structure, most relevant information is concentrated in small number of low dimensional manifolds. If a linear subspace is a flat sheet of paper, then a rolled up sheet of paper is a simple example of a nonlinear manifold. Informally, this is called a Swiss roll, a canonical problem in the field of non-linear dimensionality reduction. Some popular manifold learning methods are,

1. Multi-dimensional scaling (MDS) : A technique used for analyzing similarity or dissimilarity of data as distances in a geometric spaces. Projects data to a lower dimension such that data points that are close to each other (in terms if Euclidean distance) in the higher dimension are close in the lower dimension as well.
2. Isometric Feature Mapping (Isomap) : Projects data to a lower dimension while preserving the geodesic distance (rather than Euclidean distance as in MDS). Geodesic distance is the shortest distance between two points on a curve.
3. Locally Linear Embedding (LLE): Recovers global non-linear structure from linear fits. Each local patch of the manifold can be written as a linear, weighted sum of its neighbours given enough data.
4. Hessian Eigenmapping (HLLE): Projects data to a lower dimension while preserving the local neighbourhood like LLE but uses the Hessian operator to better achieve this result and hence the name.
5. Spectral Embedding (Laplacian Eigenmaps): Uses spectral techniques to perform dimensionality reduction by mapping nearby inputs to nearby outputs. It preserves locality rather than local linearity
6. t-distributed Stochastic Neighbor Embedding (t-SNE): Computes the probability that pairs of data points in the high-dimensional space are related and then chooses a low-dimensional embedding which produce a similar distribution.

feature selection techniques

1. Filter Methods
2. Wrapper Methods and
3. Embedded Methods.

### Filter Methods

Filter Methods considers the relationship between features and the target variable to compute the importance of features.

#### F Test

F Test is a statistical test used to compare between models and check if the difference is significant between the model.

F-Test does a hypothesis testing model **X** and **Y** where **X** is a model created by just a constant and **Y** is the model created by a constant and a feature.

The least square errors in both the models are compared and checks if the difference in errors between model **X** and **Y** are significant or introduced by chance.

Scikit learn provides the **Selecting K best** features using F-Test.

sklearn.feature\_selection.f\_regression

For Classification tasks

sklearn.feature\_selection.f\_classif

F-Test checks for and only captures linear relationships between features and labels. A highly correlated feature is given higher score and less correlated features are given lower score.

1. Correlation is highly deceptive as it doesn’t capture strong non-linear relationships.
2. Using summary statistics like correlation may be a bad idea

#### Mutual Information

Mutual Information between two variables measures the dependence of one variable to another. If **X** and **Y** are two variables, and

1. If **X** and **Y** are independent, then no information about **Y** can be obtained by knowing **X** or vice versa. Hence their mutual information is **0**.
2. If **X** is a deterministic function of **Y**, then we can determine **X** from **Y** and **Y** from **X** with mutual information **1**.
3. When we have Y = f(X,Z,M,N), 0 < mutual information < 1

Advantage of using mutual information over F-Test is, it does well with the non-linear relationship between feature and target variable.

Sklearn offers feature selection with Mutual Information for regression and classification tasks.

sklearn.feature\_selection.mututal\_info\_regression   
sklearn.feature\_selection.mututal\_info\_classif

#### Variance Threshold

This method removes features with variation below a certain cutoff.

The idea is when a feature doesn’t vary much within itself, it generally has very little predictive power.

sklearn.feature\_selection.VarianceThreshold

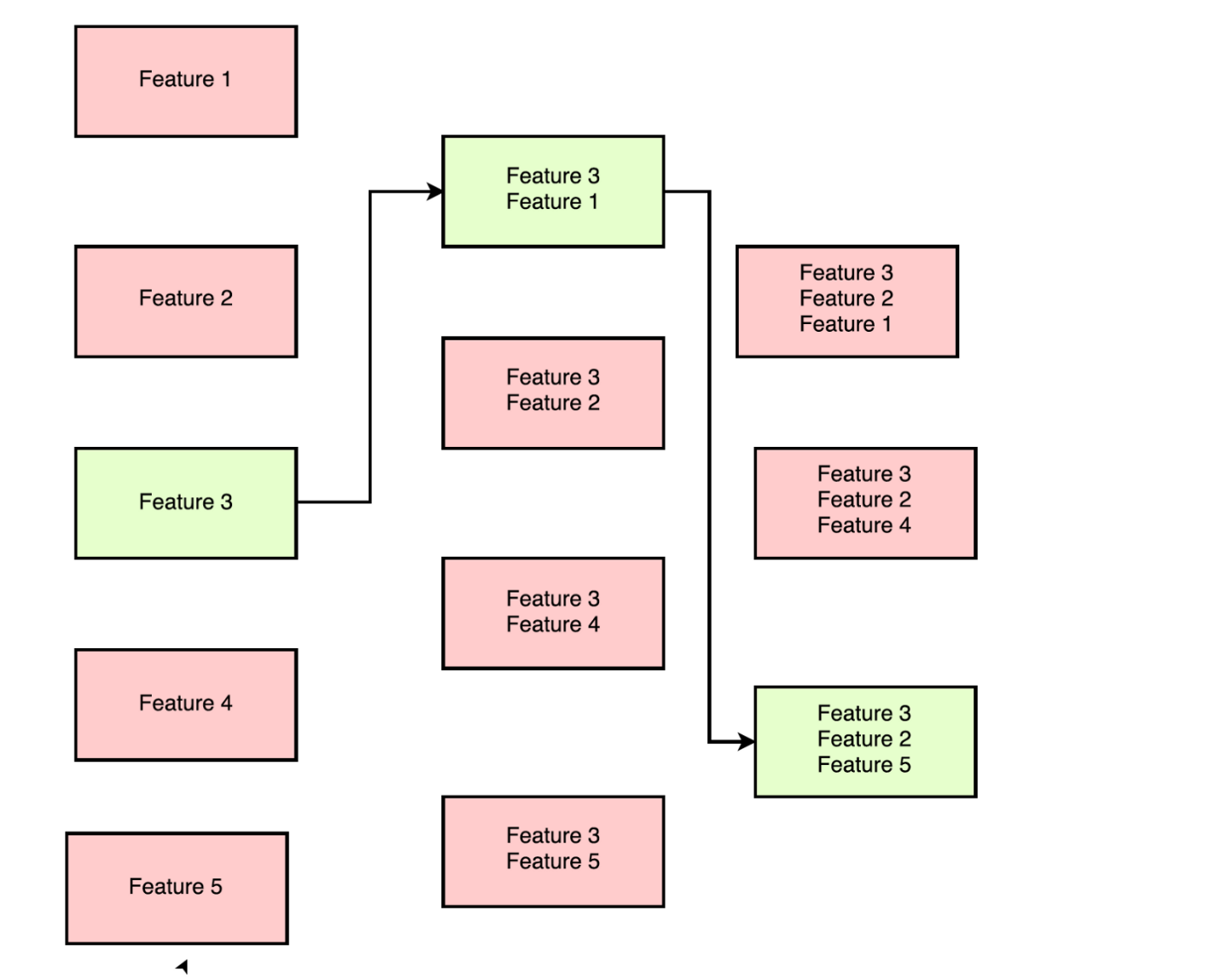
Variance Threshold doesn’t consider the relationship of features with the target variable.

### Wrapper Methods

Wrapper Methods generate models with a subsets of feature and gauge their model performances.

#### Forward Search

This method allows you to search for the best feature w.r.t model performance and add them to your feature subset one after the other.



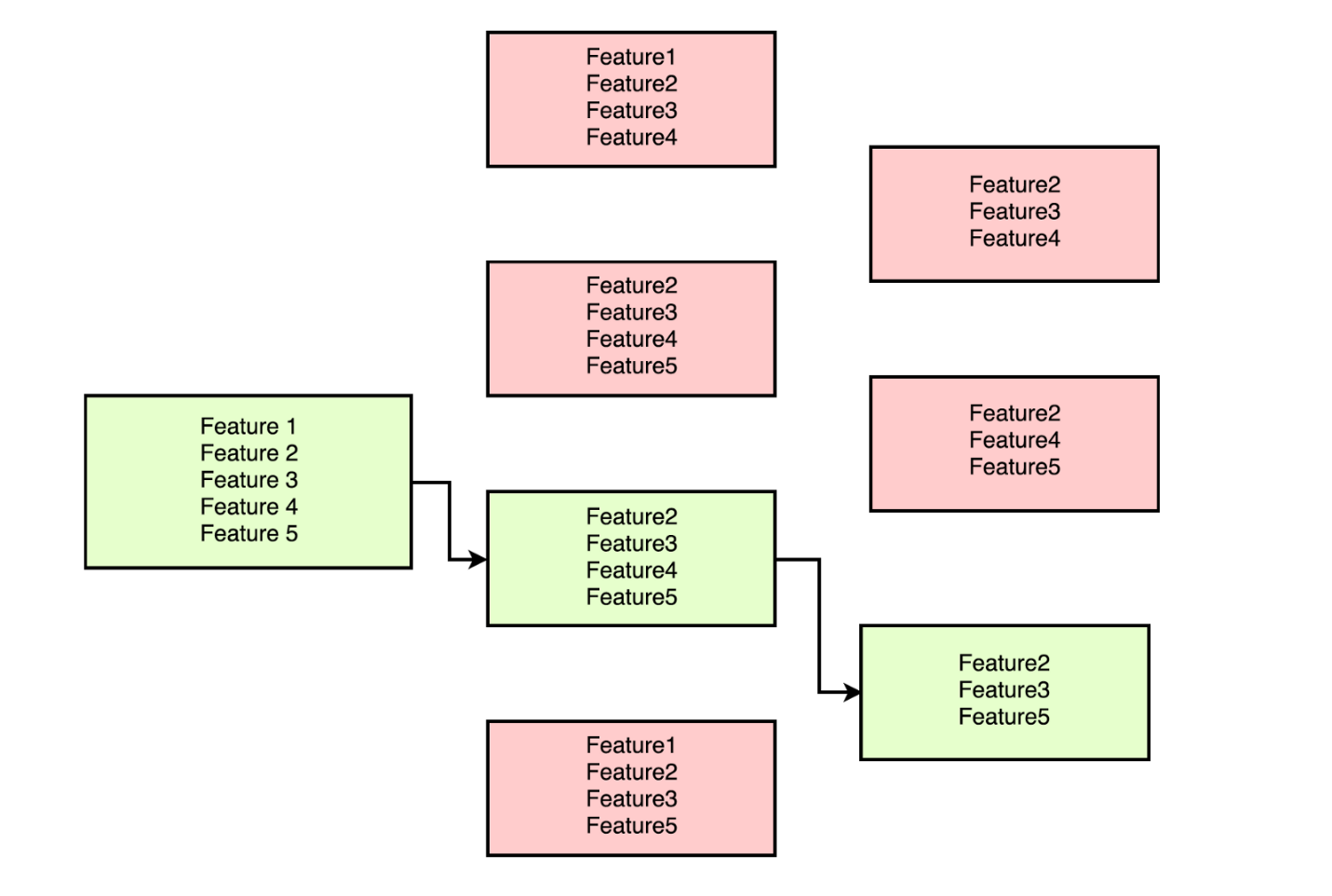
For data with n features,

->On first round ‘n’ models are created with individual feature and the best predictive feature is selected.

->On second round, ‘n-1’ models are created with each feature and the previously selected feature.

->This is repeated till a best subset of ‘m’ features are selected.

**Recursive Feature Elimination**



For data with n features,

->On first round ‘n-1’ models are created with combination of all features except one. The least performing feature is removed

-> On second round ‘n-2’ models are created by removing another feature.

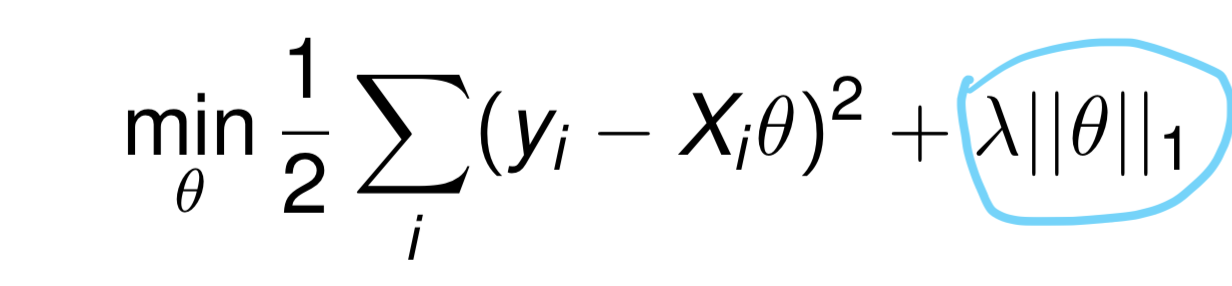
Wrapper Methods promises you a best set of features with a extensive greedy search.

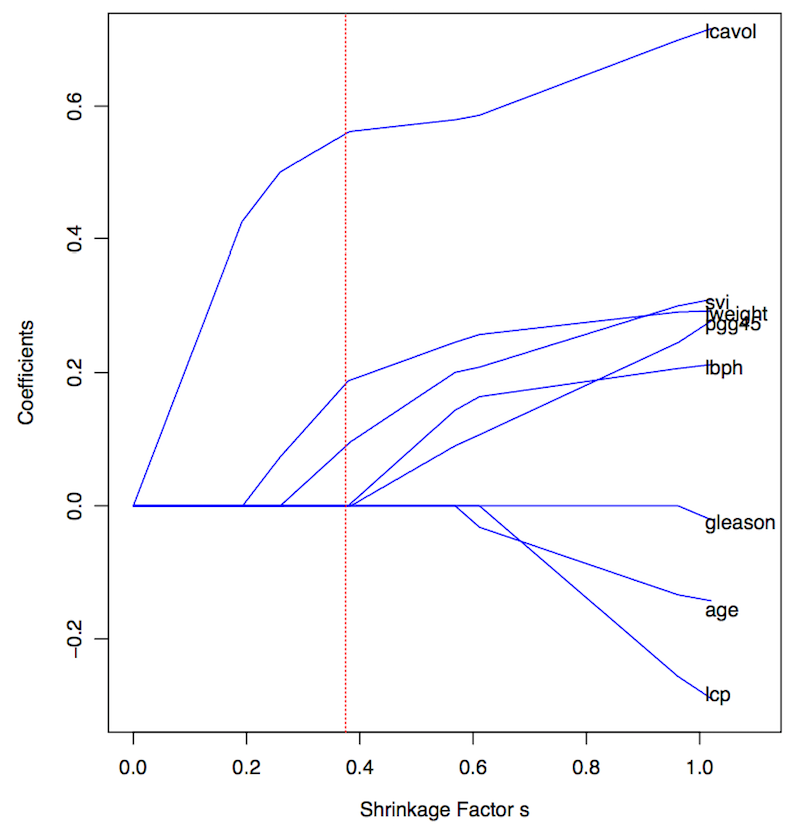
But the main drawbacks of wrapper methods is the sheer amount of models that needs to be trained. It is computationally very expensive and is infeasible with large number of features.

### **Embedded Methods**

Feature selection can also be acheived by the insights provided by some Machine Learning models.

**LASSO Linear Regression** can be used for feature selections. Lasso Regression is performed by adding an extra term to the cost function of Linear Regression. This apart from preventing overfitting also reduces the coefficients of less important features to zero.





As we vary ƛ in the cost function, the coefficients have been plotted in this graph. We observe that for ƛ ~=0, the coefficients of most of the features side towards zero. In the above graph, we can see that only ‘lcavol’, ’svi’, and ‘lweight’ are the features with non-zero coefficients when ƛ = 0.4.

**Tree based models** calculates feature importance for they need to keep the best performing features as close to the root of the tree. Constructing a decision tree involves calculating the best predictive feature.

The feature importance in tree-based models are calculated based on **Gini Index**, **Entropy** or **Chi-Square** value.