**Find below some techniques for dimensionality reduction I find very useful:**

1. **Principal Component Analysis:**

[Principal Component Analysis (PCA)](http://en.wikipedia.org/wiki/Principal_component_analysis) is a statistical procedure that orthogonally transforms the original n coordinates of a data set into a new set of n coordinates called principal components. As a result of the transformation, the first principal component has the largest possible [variance](http://en.wikipedia.org/wiki/Variance); each succeeding component has the highest possible variance under the constraint that it is [orthogonal](http://en.wikipedia.org/wiki/Orthogonal) to (i.e., uncorrelated with) the preceding components. Keeping only the first m < ncomponents reduces the data dimensionality while retaining most of the data information, i.e. the variation in the data. Notice that the PCA transformation is sensitive to the relative scaling of the original variables. Data column ranges need to be normalized before applying PCA. Also notice that the new coordinates (PCs) are not real system-produced variables anymore. Applying PCA to your data set loses its interpretability. If interpretability of the results is important for your analysis, PCA is not the transformation for your project.

See an example of PCA in R in Feature Engineering folder. ☺

1. **Backward Feature Elimination**.

In this technique, at a given iteration, the selected classification algorithm is trained on *n* input features. Then we remove one input feature at a time and train the same model on *n-1* input features *n* times. The input feature whose removal has produced the smallest increase in the error rate is removed, leaving us with *n-1* input features. The classification is then repeated using *n-2* features, and so on. Each iteration *k* produces a model trained on *n-k* features and an error rate *e(k).*Selecting the maximum tolerable error rate, we define the smallest number of features necessary to reach that classification performance with the selected machine learning algorithm.

1. **Forward Feature Construction**.

This is the inverse process to the Backward Feature Elimination. We start with 1 feature only, progressively adding 1 feature at a time, i.e. the feature that produces the highest increase in performance. Both algorithms, Backward Feature Elimination and Forward Feature Construction, are quite time and computationally expensive. They are practically only applicable to a data set with an already relatively low number of input columns.

There are many more techniques which can be useful for feature selection and engineering. Please refer the link below for more information:

<http://www.kdnuggets.com/2015/05/7-methods-data-dimensionality-reduction.html>