



We gave participants the challenge to identify **Human Activity Recognition Using Smartphones Data Set** (<https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones>). The data set had **561 variables** for training model used for the identification of Human activity in test data set.

### Human Activity Recognition Using Smartphones Data Set

Download [Data Folder](#) [Data Set Description](#)

Abstract: Human Activity Recognition database built from the recordings of 30 subjects performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors.

Data Set Characteristics:	Multivariate, Time-Series	Number of Instances:	10299	Area:	Computer
Attribute Characteristics:	N/A	Number of Attributes:	561	Date Donated	2012-12-10
Associated Tasks:	Classification, Clustering	Missing Values?	N/A	Number of Web Hits:	206652

([https://www.analyticsvidhya.com/wp-content/uploads/2015/07/Human\\_Data.png](https://www.analyticsvidhya.com/wp-content/uploads/2015/07/Human_Data.png))

The participants in hackathon had varied experience and expertise level. As expected, the experts did a commendable job at identifying the human activity. However, beginners & intermediates struggled with sheer number of variables in the dataset (561 variables). Under the pressure of time, these people tried using variables really without understanding the significance level of variable(s). They lacked the skill to filter information from seemingly high dimensional problems and reduce them to a few relevant dimensions – the skill of dimension reduction.

Further, this lack of skill came across in several forms in way of questions asked by various participants:

- There are too many variables – do I need to explore each and every variable?
- Are all variables important?
- All variables are numeric and what if they have multi-collinearity? How can I identify these variables?
- I want to use decision tree. It can automatically select the right variables. Is this a right technique?
- I am using random forest but it is taking a high execution time because of high number of features
- Is there any machine learning algorithm that can identify the most significant variables automatically?
- As this is a classification problem, can I use SVM with all variables?
- Which is the best tool to deal with high number of variable, R or Python?

If you have faced similar questions, you are reading the right article. In this article, we will look at various methods to identify the significant variables using the most common **dimension reduction techniques and methods**.

### Table of Contents

We use cookies on Analytics Vidhya websites to deliver our services, analyze web traffic, and improve your experience on the site. By using

Analytics Vidhya, you agree to our Privacy Policy (<https://www.analyticsvidhya.com/privacy-policy/>) and Terms of Use

1. Why Dimension Reduction is Important in machine learning and predictive modeling? (<https://www.analyticsvidhya.com/terms/>)

Accept

2. What are Dimension Reduction techniques?
3. What are the benefits of using Dimension Reduction techniques?
4. What are the common methods to reduce number of Dimensions?
5. Is Dimensionality Reduction good or bad?

## Why Dimension Reduction is important in machine learning & predictive modeling?

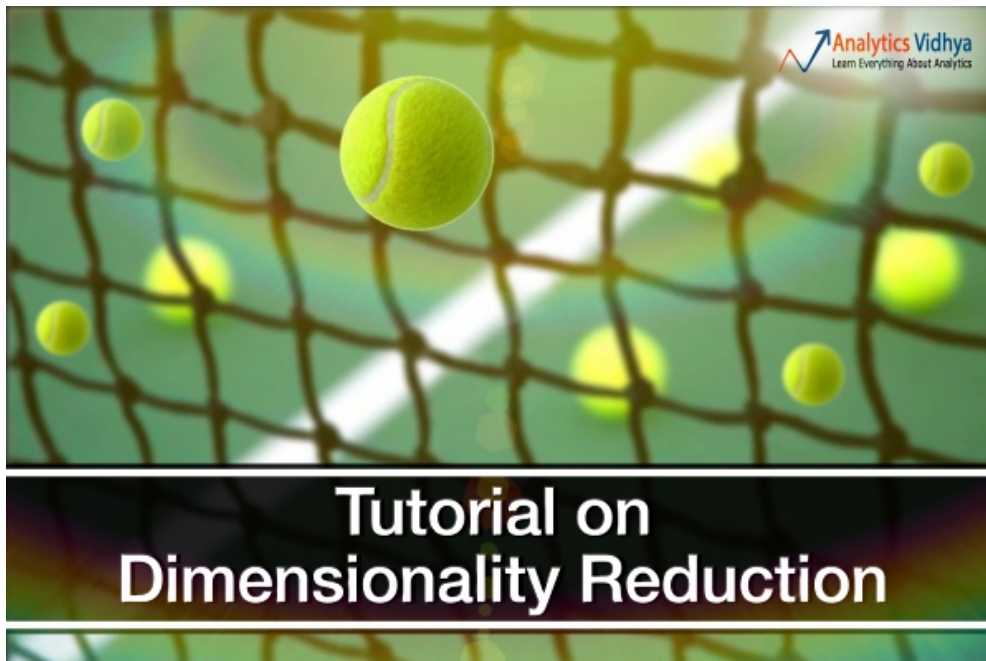
The problem of unwanted increase in dimension is closely related to fixation of measuring / recording data at a far granular level than it was done in past. This is no way suggesting that this is a recent problem. It has started gaining more importance lately due to surge in data.

Lately, there has been a tremendous increase in the way sensors are being used in the industry. These sensors continuously record data and store it for analysis at a later point. In the way data gets captured, there can be a lot of redundancy. For example, let us take case of a motorbike rider in racing competitions. Today, his position and movement gets measured by GPS sensor on bike, gyro meters, multiple video feeds and his smart watch. Because of respective errors in recording, the data would not be exactly same. However, there is very little incremental information on position gained from putting these additional sources. Now assume that an analyst sits with all this data to analyze the racing strategy of the biker – he/ she would have a lot of variables / dimensions which are similar and of little (or no) incremental value. This is the problem of high unwanted dimensions and needs a treatment of dimension reduction.

Let's look at other examples of new ways of data collection:

- Casinos are capturing data using cameras and tracking each and every move of their customers.
- Political parties are capturing data by expanding their reach on field
- Your smart phone apps collect a lot of personal details about you
- Your set top box collects data about which programs preferences and timings
- Organizations are evaluating their brand value by social media engagements (comments, likes), followers, positive and negative sentiments

With more variables, comes more trouble! And to avoid this trouble, **dimension reduction techniques** come to the rescue.

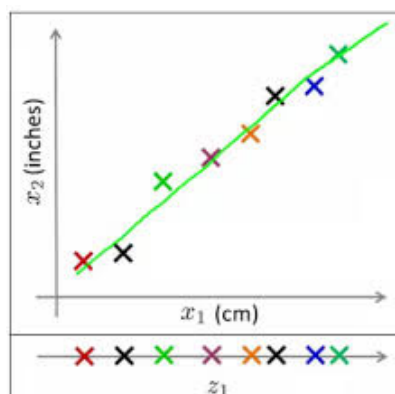


(<https://www.analyticsvidhya.com/wp-content/uploads/2015/07/tennis-final-00051.jpg>)

## What are Dimension Reduction techniques?

Dimension Reduction refers to the process of converting a set of data having vast dimensions into data with lesser dimensions ensuring that it conveys similar information concisely. These techniques are typically used while solving **machine learning problems** to obtain better features for a classification or regression task.

Let's look at the image shown below. It shows 2 dimensions  $x_1$  and  $x_2$ , which are let us say measurements of several object in cm ( $x_1$ ) and inches ( $x_2$ ). Now, if you were to use both these dimensions in machine learning, they will convey similar information and introduce a lot of noise in system, so you are better off just using one dimension. Here we have converted the dimension of data from 2D (from  $x_1$  and  $x_2$ ) to 1D ( $z_1$ ), which has made the data relatively easier to explain.



([https://www.analyticsvidhya.com/wp-content/uploads/2015/07/2D\\_1D.jpg](https://www.analyticsvidhya.com/wp-content/uploads/2015/07/2D_1D.jpg))

In similar ways, we can reduce  $n$  dimensions of data set to  $k$  dimensions ( $k < n$ ). These  $k$  dimensions can be directly identified (filtered) or can be a combination of dimensions (weighted averages of dimensions) or new dimension(s) that represent existing multiple dimensions well.

We use cookies on Analytics Vidhya websites to deliver our services, analyze web traffic, and improve your experience on the site. By using

Analytics Vidhya, you agree to our Privacy Policy (<https://www.analyticsvidhya.com/privacy-policy/>) and Terms of Use

(<https://www.analyticsvidhya.com/terms/>). **Accept**

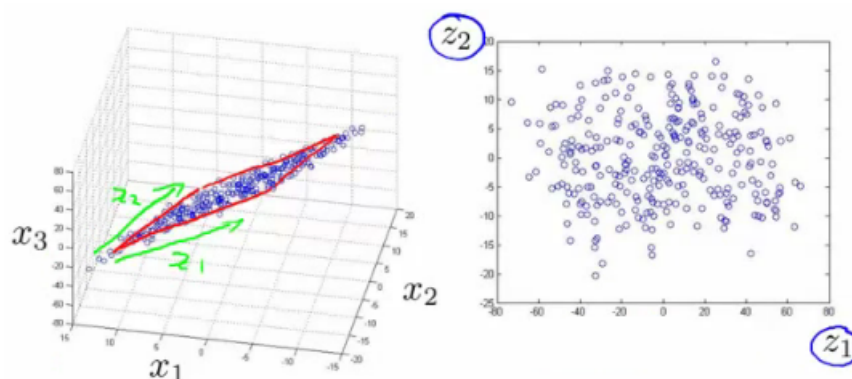
One of the most common application of this technique is **Image processing**. You might have come across this Facebook application – “**Which Celebrity Do You Look Like?** (<https://apps.facebook.com/qbquiz-fntbv/>)”. But, have you ever thought about the algorithm used behind this?

Here's the answer: To identify the matched celebrity image, we use pixel data and each pixel is equivalent to one dimension. In every image, there are high number of pixels i.e. high number of dimensions. And every dimension is important here. You can't omit dimensions randomly to make better sense of your overall data set. In such cases, dimension reduction techniques help you to find the significant dimension(s) using various method(s). We'll discuss these methods shortly.

## What are the benefits of Dimension Reduction?

Let's look at the benefits of applying Dimension Reduction process:

- It helps in data compressing and reducing the storage space required
- It fastens the time required for performing same computations. Less dimensions leads to less computing, also less dimensions can allow usage of algorithms unfit for a large number of dimensions
- It takes care of multi-collinearity that improves the model performance. It removes redundant features. For example: there is no point in storing a value in two different units (meters and inches).
- Reducing the dimensions of data to 2D or 3D may allow us to plot and visualize it precisely. You can then observe patterns more clearly. Below you can see that, how a 3D data is converted into 2D. First it has identified the 2D plane then represented the points on these two new axis  $z_1$  and  $z_2$ .



(<https://www.analyticsvidhya.com/wp-content/uploads/2015/07/Image-4.png>)

- It is helpful in noise removal also and as result of that we can improve the performance of models.

## What are the common methods to perform Dimension Reduction?

There are many methods to perform Dimension reduction. I have listed the most common methods below:

**1. Missing Values:** (<https://www.analyticsvidhya.com/blog/2015/02/7-steps-data-exploration-preparation-building-model-part-2/>) While exploring data, if we encounter missing values, what we do? Our first step should be to identify the reason for the missing values and then drop the variables using appropriate methods. But what if we have too many missing values? Should we drop the missing values or drop the variables? (<https://www.analyticsvidhya.com/blog/2015/02/7-steps-data-exploration-preparation-building-model-part-2/>) or drop the variables? (<https://www.analyticsvidhya.com/terms/>). **Accept**

I would prefer the latter, because it would not have lot more details about data set. Also, it would not help in improving the power of model. Next question, is there any threshold of missing values for dropping a variable? It varies from case to case. If the information contained in the variable is not that much, you can drop the variable if it has more than ~40-50% missing values.

**2. Low Variance:** Let's think of a scenario where we have a constant variable (all observations have same value, 5) in our data set. Do you think, it can improve the power of model? Ofcourse NOT, because it has zero variance. In case of high number of dimensions, we should drop variables having low variance compared to others because these variables will not explain the variation in target variables.

**3. Decision Trees:** (<https://www.analyticsvidhya.com/blog/2015/01/decision-tree-simplified/>) It is one of my favorite techniques. It can be used as a ultimate solution to tackle multiple challenges like missing values, outliers and identifying significant variables. It worked well in our Data Hackathon also. Several data scientists used decision tree and it worked well for them.

**4. Random Forest:** (<https://www.analyticsvidhya.com/blog/2014/06/introduction-random-forest-simplified/>) Similar to decision tree is Random Forest. I would also recommend using the in-built feature importance provided by random forests to select a smaller subset of input features. Just be careful that random forests have a tendency to bias towards variables that have more no. of distinct values i.e. favor numeric variables over binary/categorical values.

**5. High Correlation:** Dimensions exhibiting higher correlation can lower down the performance of model. Moreover, it is not good to have multiple variables of similar information or variation also known as "**Multicollinearity**". You can use *Pearson* (continuous variables) or *Polychoric* (discrete variables) correlation matrix to identify the variables with high correlation and select one of them using VIF ([https://en.wikipedia.org/wiki/Variance\\_inflation\\_factor](https://en.wikipedia.org/wiki/Variance_inflation_factor)) (Variance Inflation Factor). Variables having higher value (  $VIF > 5$  ) can be dropped.

**6. Backward Feature Elimination:** In this method, we start with all  $n$  dimensions. Compute the sum of square of error (SSR) after eliminating each variable ( $n$  times). Then, identifying variables whose removal has produced the smallest increase in the SSR and removing it finally, leaving us with  $n-1$  input features.

Repeat this process until no other variables can be dropped. Recently in Online Hackathon organised by Analytics Vidhya (11-12 Jun'15) (<http://discuss.analyticsvidhya.com/t/data-hackathon-online-11th-12th-july-2015-register-here/1599>), Data scientist who held second position used Backward Feature Elimination in linear regression to train his model.

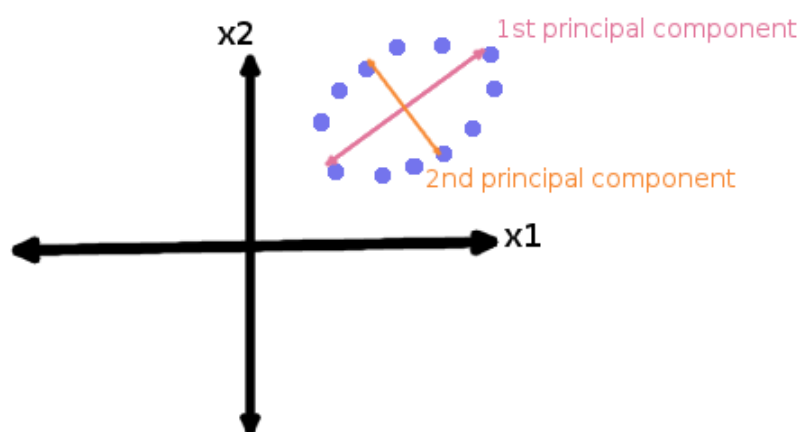
Reverse to this, we can use "**Forward Feature Selection**" method. In this method, we select one variable and analyse the performance of model by adding another variable. Here, selection of variable is based on higher improvement in model performance.

**7. Factor Analysis:** Let's say some variables are highly correlated. These variables can be grouped by their correlations i.e. all variables in a particular group can be highly correlated among themselves but have low correlation with variables of other group(s). Here each group represents a single underlying construct or factor. These factors are small in number as compared to large number of dimensions. However, these factors are difficult to observe. There are basically two methods of performing factor analysis:

- EFA (Exploratory Factor Analysis)
- CFA (Confirmatory Factor Analysis)

**8. Principal Component Analysis (PCA):** In this technique, variables are transformed into a new set of variables, which are linear combination of original variables. These new set of variables are known as **principle components**. They are obtained in such a way that first principle component accounts for most of the possible variation of original data after which each succeeding component has the highest possible variance.

The second principal component must be orthogonal to the first principal component. In other words, it does its best to capture the variance in the data that is not captured by the first principal component. For two-dimensional dataset, there can be only two principal components. Below is a snapshot of the data and its first and second principal components. You can notice that second principle component is orthogonal to first principle component.



(<https://www.analyticsvidhya.com/wp-content/uploads/2015/07/PCA.png>) The principal components are sensitive to the scale of measurement, now to fix this issue we should always standardize variables before applying PCA. Applying PCA to your data set loses its meaning. If interpretability of the results is important for your analysis, PCA is not the right technique for your project.

## Is Dimension Reduction Good or Bad?

Recently, we received this question on our data science forum. Here's the complete answer (<http://discuss.analyticsvidhya.com/t/dimensionality-reduction-is-good-or-bad/2444/4>).

We use cookies on Analytics Vidhya websites to deliver our services, analyze web traffic, and improve your experience on the site. By using

**End Note** Analytics Vidhya, you agree to our Privacy Policy (<https://www.analyticsvidhya.com/privacy-policy/>) and Terms of Use

(<https://www.analyticsvidhya.com/terms/>).