1. What are the key reasons for reducing the dimensionality of a dataset? What are the major disadvantages?

**Key Reasons:**

* It helps in data compression, and hence reduced storage space.
* It reduces computation time.
* It also helps remove redundant features, if any.

**Disadvantages:**

* It may lead to some amount of data loss.
* PCA tends to find linear correlations between variables, which is sometimes undesirable.
* PCA fails in cases where mean and covariance are not enough to define datasets.

2. What is the dimensionality curse?

The curse of dimensionality basically means **that the error increases with the increase in the number of features**. It refers to the fact that algorithms are harder to design in high dimensions and often have a running time exponential in the dimensions.

Dimensionality reduction is a method of converting the high dimensional variables into lower dimensional variables without changing the specific information of the variables. To overcome the issue of the curse of dimensionality, Dimensionality Reduction is used to reduce the feature space with consideration by a set of principal features. Dimensionality Reduction contains no extra variables that make the data analysing easier and simple for [machine learning](https://analyticsindiamag.com/popular-machine-learning-interview-questions-used-to-assess-candidates/) algorithms and resulting in a faster outcome from the algorithms.

3. Tell if its possible to reverse the process of reducing the dimensionality of a dataset? If so, how can you go about doing it? If not, what is the reason?

NO

Data may be lost

4. Can PCA be utilized to reduce the dimensionality of a nonlinear dataset with a lot of variables?

OF course, you can still do a PCA computation on **nonlinear data** - but the results will be meaningless, beyond decomposing to the dominant linear modes and provided a global linear representation of the spread of the data.

5. Assume you're running PCA on a 1,000-dimensional dataset with a 95 percent explained variance ratio. What is the number of dimensions that the resulting dataset would have?

Depends on the dataset! If all the variance is in a few dimensions this can be very low (technically down to 1 dimension). If the instances of original dataset are uniformly distributed, the reduced dataset will have many dimensions (technically up to 95% of the number of dimensions as the original dataset).

6. Will you use vanilla PCA, incremental PCA, randomized PCA, or kernel PCA in which situations?

* Vanilla PCA: the dataset fit in memory
* Incremental PCA: larget dataset that don't fit in memory, online taks
* Randomized PCA: considerably reduce dimensionality and the dataset fit the memory.
* kenrl PCA: used for nonlinear PCA

7. How do you assess a dimensionality reduction algorithm's success on your dataset?

* measure the reconstruction error
* measure the performance in second Machine Learning algorithm

8. Is it logical to use two different dimensionality reduction algorithms in a chain?

YES. One can use a fast projection method (PCA) to first get rid of useless dimensions (i.e. dimensions that have no variance), and then use a slow manifold learning methods (LLE) to ‘unfold’ then remaining dataset to even lower dimensions.