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

disadvantages?

Dimensionality reduction brings many advantages to your machine learning data, including: Fewer features mean less complexity. You will need less storage space because you have fewer data. Fewer features require less computation time

2. What is the dimensionality curse?

The curse of dimensionality refers to various phenomena that arise when analyzing and organizing data in high-dimensional spaces that do not occur in low-dimensional settings such as the three-dimensional physical space of everyday experience

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?

Space required to store the data is reduced as the number of dimensions comes down

Less dimensions lead to less computation/training time

Some algorithms do not perform well when we have a large dimensions. So reducing these dimensions needs to happen for the algorithm to be useful

It takes care of multicollinearity by removing redundant features. For example, you have two variables – ‘time spent on treadmill in minutes’ and ‘calories burnt’. These variables are highly correlated as the more time you spend running on a treadmill, the more calories you will burn. Hence, there is no point in storing both as just one of them does what you require

It helps in visualizing data. As discussed earlier, it is very difficult to visualize data in higher dimensions so reducing our space to 2D or 3D may allow us to plot and observe patterns more clearly

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

Yes, PCA can be used to significantly reduce dimensionality of highly non-linear dataset because it can at least get rid of useless dimensions. If there are no useless dimensions, reducing dimensionality with PCA will lose too much information.

5. Assume youe 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?

float: If 0 < n\_components < 1, PCA will select the number of components such that the amount of variance that needs to be explained². For example, if n\_components=0.95, the algorithm will select the number of components while preserving 95% of the variance in the data.

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

Regular PCA is the default, but it works only if the dataset fits in memory. Incremental PCA is useful for large datasets that don't fit in memory, but it is slower than regular PCA, so if the dataset fits in memory you should prefer regular PCA

7. How do you assess a dimensionality reduction algorithms success on your dataset?

A dimensionality reduction algorithm is said to work well if it eliminates a significant number of dimensions from the dataset without losing too much information. Moreover, the use of dimensionality reduction in preprocessing before training the model allows measuring the performance of the second algorithm.

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

It can absolutely make sense to chain two different dimensionality reduction algorithms. A common example is using PCA to quickly get rid of a large number of useless dimensions, then applying another much slower dimensionality reduction algorithm, such as LLE.