UnCodemy

Dimensionality Reduction and PCA PDF

Dimensionality reduction cuts the number of features while keeping most useful information. Principal component analysis, or PCA, builds new uncorrelated features called components, ordered by how much variance they capture. Scale the data first, choose the number of components by explained variance, and remember that components are harder to interpret than the original columns.

Key takeaways

- Reduction helps with visualisation, speed, noise and the curse of dimensionality.
- PCA needs scaled numeric features, or large-scale columns will dominate.
- The explained variance ratio shows how much information each component keeps.
- PCA is linear and unsupervised, so it ignores the target.
- t-SNE and UMAP are mainly for visualisation, not for general modelling input.

What is dimensionality reduction and PCA?

Dimensionality reduction is a set of techniques that transform data with many features into fewer features while keeping as much structure as possible, and PCA does this using directions of greatest variance.

Why reduce dimensions at all?

Datasets can have hundreds of columns, many of them correlated or noisy. Too many features slow training, increase the risk of overfitting and make plots impossible. In very high dimensions, distances between points become less meaningful, a problem known as the **curse of dimensionality**. Reduction can also compress images and remove redundant sensor signals.

How does PCA work in simple words?

PCA finds new axes through the data. The first axis points where the data varies most, the second axis is perpendicular to it and captures the next largest variation, and so on. Each data point is then described by its position along the first few axes. These axes are called **principal components** and they are combinations of the original features.

What steps do I follow?

1. Select numeric features and handle missing values.
2. **Standardise** each feature to mean zero and unit spread.

3. Fit PCA on training data only.
4. Inspect the explained variance ratio.
5. Choose the number of components, for example enough to keep around ninety percent of variance, depending on the task.
6. Transform training and test data with the same fitted PCA.
7. Check the model score with and without reduction.

How do I read explained variance?

```
component   explained_variance_ratio   cumulative
PC1         0.46                       0.46
PC2         0.23                       0.69
PC3         0.12                       0.81
PC4         0.08                       0.89
```

The cumulative column tells you how much of the original variation the first few components retain. A sharp bend in a scree plot, where gains flatten, is a common hint for the cut-off.

Can I show a minimal code pattern?

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_train)
pca = PCA(n_components=0.90)
X_pca = pca.fit_transform(X_scaled)
print(pca.explained_variance_ratio_)
```

Passing a fraction asks scikit-learn to keep enough components to explain that share of variance. Place scaling and PCA inside a Pipeline so the test data uses the same fitted steps.

What are the limits of PCA?

PCA captures only linear structure and maximises variance, which is not always what predicts the target. Components can be hard to explain to stakeholders, since each mixes many original columns. Outliers can distort the axes. If interpretability matters, consider feature selection instead of extraction.

What about t-SNE and UMAP?

These methods map high-dimensional data to two or three dimensions for plots, keeping local neighbourhoods together. They are good for exploring clusters in embeddings or images. Their distances between far-apart groups are not reliable, results vary with settings and they are not normally used as input features for production models.

When should I avoid reduction?

Skip it when there are few features, when features are already interpretable and important, or when a tree-based model handles many columns well. Always compare against a baseline to confirm that reduction helps.

How can PCA help with visualisation?

Projecting data onto the first two components lets you draw a scatter plot of a dataset that has dozens of columns. Colour the points by class and look for visible groups or overlap. If classes separate in the plot, a simple model may work well; if they overlap heavily, you may need better features. Remember that a two-component plot may hide structure that lies in later components, so check the explained variance before drawing conclusions from the picture.

What alternatives exist for reducing features?

Feature selection keeps a subset of the original columns using correlation filters, model-based importance or recursive elimination, and it preserves interpretability. Regularised models such as lasso shrink unhelpful coefficients to zero. Autoencoders, a neural method, learn compressed representations for images and other complex data. Choose the method by your goal: use PCA for decorrelation and compression, selection for explainable models and autoencoders when you have enough data and a nonlinear structure.

How do I explain PCA to a non-technical audience?

Say that PCA rotates the view of the data to find the angles that show the most difference between rows, then keeps only the most informative angles. A useful comparison is photographing a tall building: you pick the viewpoint that shows the most detail and ignore the others. Add one sentence on the trade-off, which is that you gain simplicity and speed but lose some detail and the direct meaning of each original column. This plain framing helps managers accept the result and ask better questions.

Frequently asked questions

What does PCA do?

PCA creates new uncorrelated features called principal components, ordered by the variance they capture, so you can keep fewer features with most of the information.

Do I need to scale data before PCA?

Yes. Standardise features first, otherwise columns with large numeric ranges dominate the components and distort the result.

How many components should I keep?

Choose using the cumulative explained variance, a scree plot and the effect on model performance, rather than a fixed number.

Is PCA supervised?

No. PCA ignores the target and looks only at feature variance, so components with high variance may not be the most predictive.

Can t-SNE be used for modelling?

It is mainly for visualisation. Distances in a t-SNE plot are not reliable for modelling, so use it for exploration only.

Sources and further reading

- scikit-learn user guide
- scikit-learn decomposition guide

About this PDF

These PDFs are study material for learning and revision. They are not a certificate, a degree or a promise of a job, and technology changes, so check official documentation for the latest details. Spot an error? Write to info@uncodemy.com and the trainer team will review it.

More free resources and courses: uncodemy.com | info@uncodemy.com | +91 98183 66550