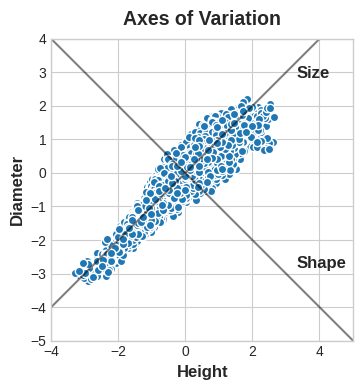
Feature Engineering: 5th lesson – Principal Component Analysis

In the previous lesson, we looked at our first model-based method for feature engineering: clustering. In this lesson we look at our next: principal component analysis (PCA). Just like clustering is a partitioning of the dataset based on proximity, you could think of PCA as a partitioning of the variation in the data. PCA is a great tool to help you discover important relationships in the data and can also be used to create more informative features.

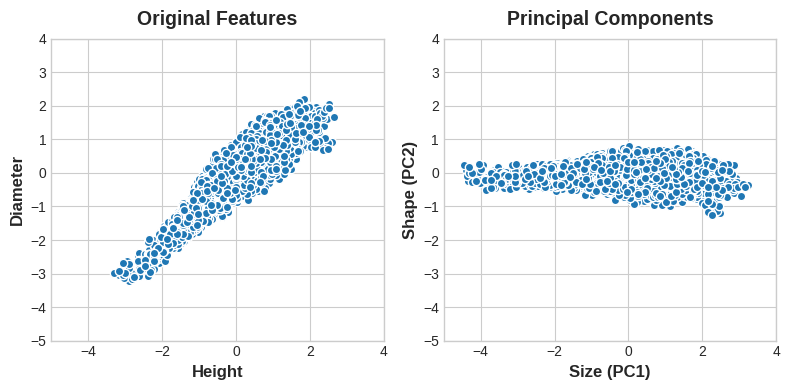
Principal component analysis:

In the *Abalone* dataset are physical measurements taken from several thousand Tasmanian abalone (an abalone is a sea creature much like a clam or an oyster). We'll just look at a couple features for now: the 'Height' and 'Diameter' of their shells. You could imagine that within this data are "axes of variation" that describe the ways the abalone tend to differ from one another. Pictorially, these axes appear as perpendicular lines running along the natural dimensions of the data, one axis for each original feature.



Often, we can give names to these axes of variation. The longer axis we might call the "Size" component: small height and small diameter (lower left) contrasted with large height and large diameter (upper right). The shorter axis we might call the "Shape" component: small height and large diameter (flat shape) contrasted with large height and small diameter (round shape).

Notice that instead of describing abalones by their 'Height' and 'Diameter', we could just as well describe them by their 'Size' and 'Shape'. This, in fact, is the whole idea of PCA: instead of describing the data with the original features, we describe it with its axes of variation. The axes of variation become the new features.



The new features PCA constructs are actually just linear combinations, or weighted sums of the original features:

df[“Size”] = 0.707 \* X[“Height”] + 0.707 \* X[“Diameter”]

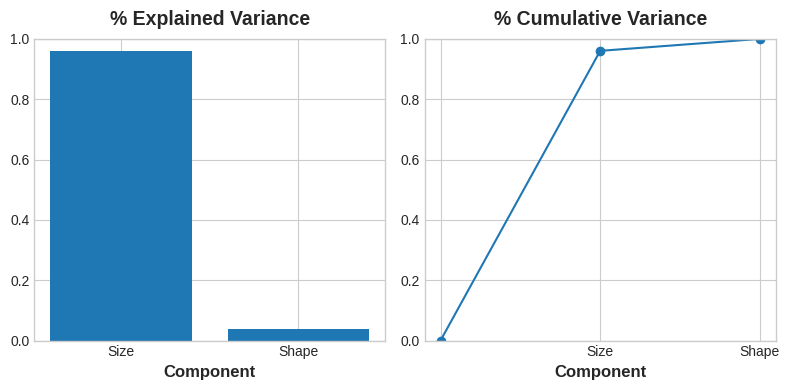
df[“Shape”] = 0.707 \* X[“Height”] - 0.707 \* X[“Diameter”]

These new features are called the principal components of the data. The weights themselves are called loadings. There will be as many principal components as there are features in the original dataset: if we had used ten features instead of two, we would have ended up with ten components.

A component's loadings tell us what variation it expresses through signs and magnitudes:

| Features \ Components | Size (PC1) | Shape (PC2) |
| --- | --- | --- |
| Height | 0.707 | 0.707 |
| Diameter | 0.707 | -0.707 |

This table of loadings is telling us that in the Size component, Height and Diameter vary in the same direction (same sign), but in the Shape component they vary in opposite directions (opposite sign). In each component, the loadings are all of the same magnitude and so the features contribute equally in both. PCA also tells us the amount of variation in each component. We can see from the figures that there is more variation in the data along the Size component than along the Shape component. PCA makes this precise through each component's *percent of explained variance*.



The Size component captures the majority of the variation between Height and Diameter. It's important to remember, however, that the amount of variance in a component doesn't necessarily correspond to how good it is as a predictor: it depends on what you're trying to predict.

PCA for feature engineering:

There are two ways you could use PCA for feature engineering.

* The first way is to use it as a descriptive technique. Since the components tell you about the variation, you could compute the MI scores for the components and see what kind of variation is most predictive of your target. That could give you ideas for kinds of features to create -- a product of Height and Diameter if Size is important, say, or a ratio of Height and Diameter if Shape is important. You could even try clustering on one or more of the high-scoring components.
* The second way is to use the components themselves as features. Because the components expose the variational structure of the data directly, they can often be more informative than the original features. Here are some use-cases.
* **Dimensionality reduction**

When your features are highly redundant or *multi-collinear* specifically, PCA will partition out the redundancy into one or more near-zero variance components, which you can then drop since they will contain little or no information.

* **Anomaly detection**

Unusual variation, not apparent from the original features, will often show up in the low-variance components. These components could be highly informative in an anomaly or outlier detection task.

* **Noise reduction**

A collection of sensor readings will often share some common background noise. PCA can sometimes collect the (informative) signal into a smaller number of features while leaving the noise alone, thus boosting the signal-to-noise ratio.

* **Decorrelation**

Some machine learning algorithms struggle with highly-correlated features. PCA transforms correlated features into uncorrelated components, which could be easier for your algorithm to work with.

PCA basically gives you direct access to the correlational structure of your data. You'll no doubt come up with applications of your own!

PCA best practices:

There are a few things to keep in mind when applying PCA.

* PCA only works with numeric features, like continuous quantities or counts.
* PCA is sensitive to scale. It is good practice to standardize your data before applying PCA, unless you know you have good reason not to.
* Consider removing or constraining outliers, since they can have an undue influence on the results.