

Figure 26-3. Visualize the principal components

Key Considerations for Performing PCA

It is vital to perform mean normalization and feature scaling on the variables of features of the original dataset before implementing PCA. This is because unscaled features can have stretched and narrow distance n-dimensional space, and this has a huge consequence when finding the principal components that explain the variance of the dataset (see Figure 26-4).

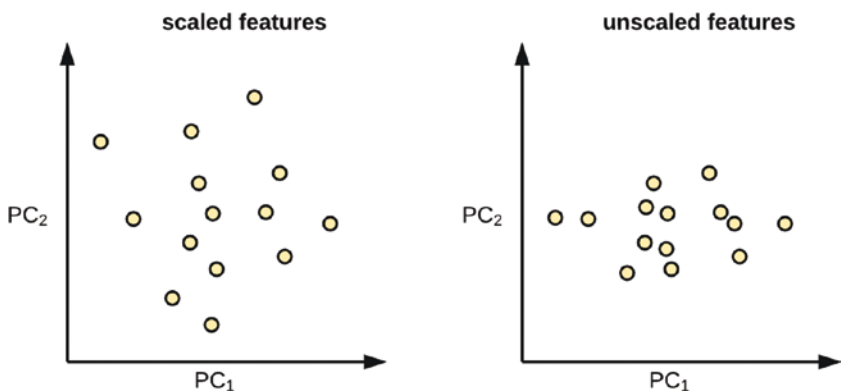


Figure 26-4. Right: An illustration of PCA with scaled features. Left: An illustration of PCA with unscaled features.

Again mean normalization ensures that every attribute or feature of the dataset has a zero mean, while feature scaling ensures all the features are within the same numeric range.

Finally, PCA is susceptible to vary wildly due to slight perturbations or changes in the dataset.

PCA with Scikit-learn

In this section, PCA is implemented using Scikit-learn.

```
# import packages
from sklearn.decomposition import PCA
from sklearn import datasets

from sklearn.preprocessing import Normalizer

# load dataset
data = datasets.load_iris()

# separate features and target
X = data.data

# normalize the dataset
scaler = Normalizer().fit(X)
normalize_X = scaler.transform(X)

# create the model.
pca = PCA(n_components=3)

# fit the model on the training set
pca.fit(normalize_X)

# examine the principal components percentage of variance explained
pca.explained_variance_ratio_

# print the principal components
pca_dataset = pca.components_
pca_dataset
'Output':
```