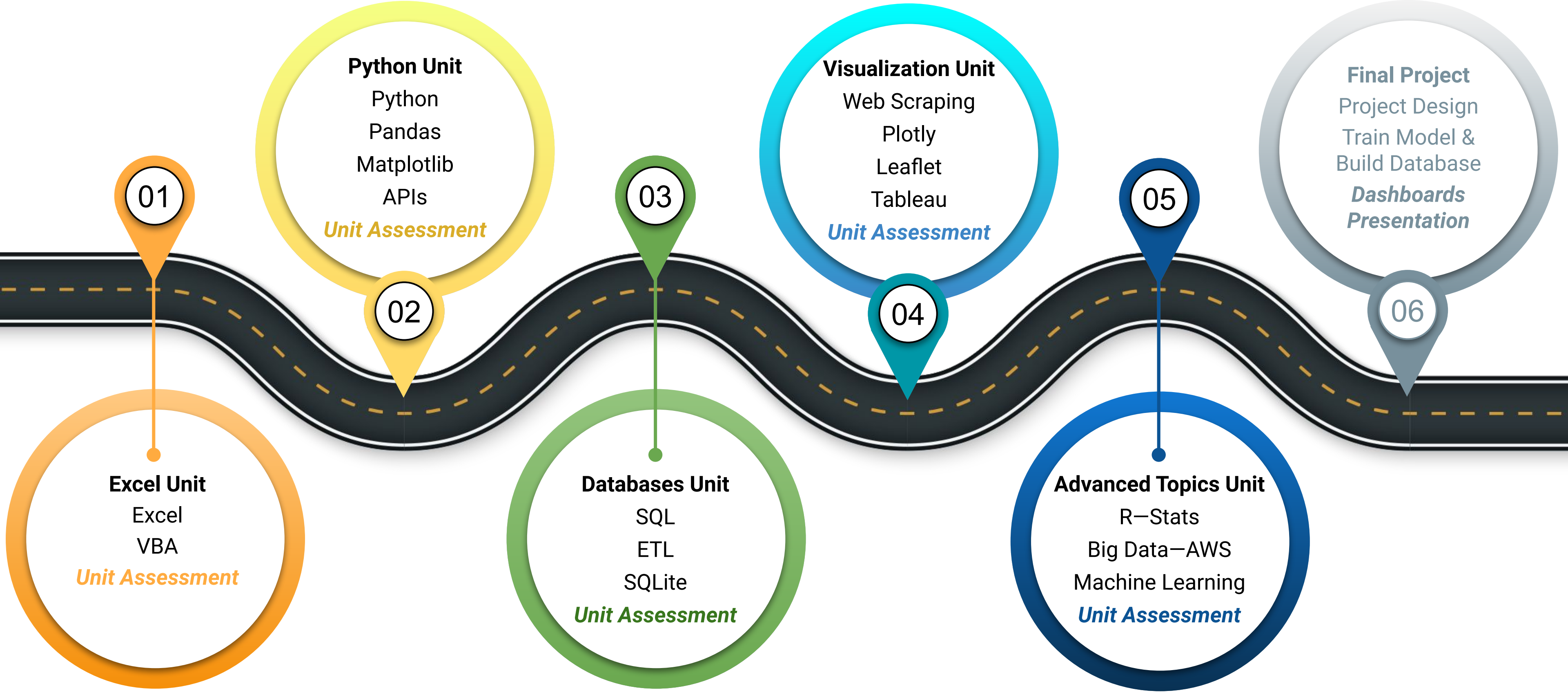
Principal Component Analysis and Hierarchical Clustering

Data Boot Camp

Lesson 18.2

The Big Picture



This Week: Unsupervised Machine Learning

By the end of this week, you'll know how to:

- To reduce features or dimensions using principal component analysis (PCA)
- Perform PCA on datasets
- Use hierarchical clustering as alternative to K-means

This Week’s Challenge

Using the skills learned throughout the week, you will analyze cryptocurrency data, reduce data dimensions with PCA, cluster with K-means, and visualize the results.

Today's Agenda

By completing today's activities, you'll learn the following skills:

01 Reducing features or dimensions using PCA

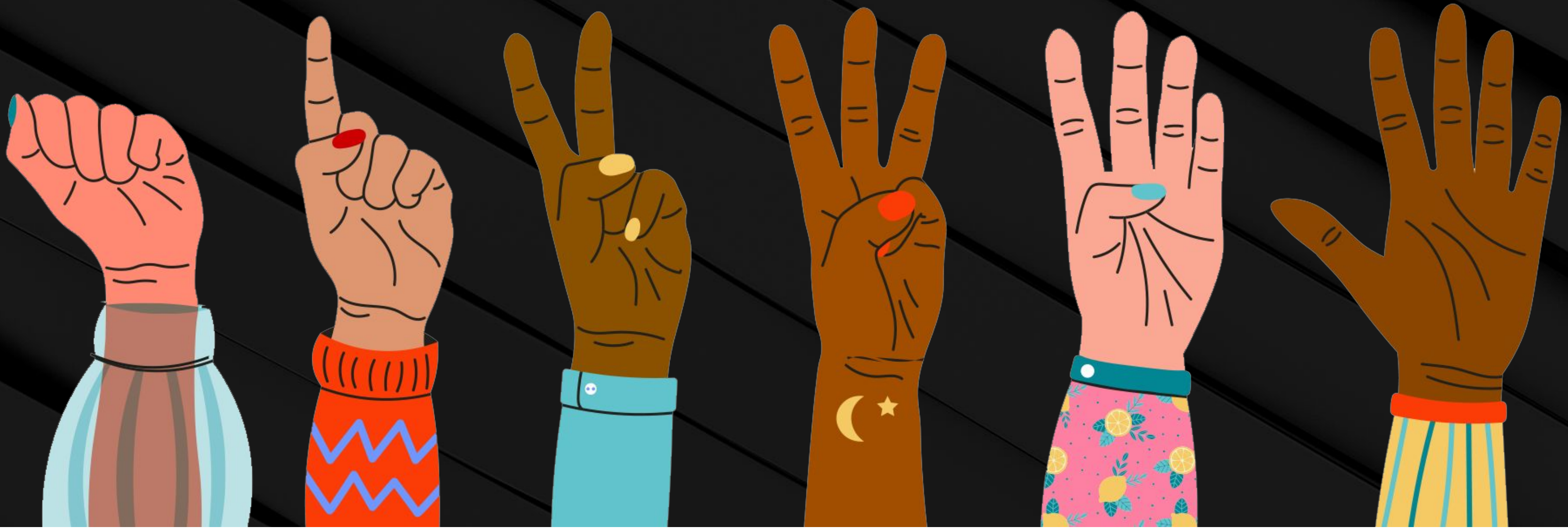
02 Performing PCA on datasets

03 Using hierarchical clustering as an alternative to K-means

Make sure you've downloaded any relevant class files!

FIST TO FIVE:

How comfortable do you feel with this topic?



Principal Component Analysis (PCA)

Principal Component Analysis

Principal component analysis is a statistical technique to speed up machine learning algorithms.

01

It does this by reducing the number of input features (or dimensions).

02

This allows us to transform large sets of variables into smaller ones that contain most of the information.

Principal Component Analysis

Before using PCA, we need to standardize the data using scikit-learn's StandardScaler module.

Then, the `fit_transform()` method combines training and transforming data into a single step.

```
# Standardize data with StandardScaler
Iris_scaled = StandardScaler().fit_transform(df_iris)
print(iris_scaled[0.5])≈
```

```
[[-0.90068117   1.03205722  -1.34127244  -1.32197673]
 [-1.14301791  -0.1249576   -1.34127244  -1.32197673]
 [-1.38535265   0.33784833  -1.39813811  -1.32197673]
 [-1.50652052   0.10644536  -1.2844067   -1.32197673]
 [-1.021849041  1.26346019  -1.3412724   -1.32197673]]
```

Principal Component Analysis

Once the features are standardized, PCA can be used to reduce the number of features.

The `n_components` parameter specifies the final number of features.

```
pca = PCA(n_components=2)
```

Principal Component Analysis

PCA reduces the dataset into a smaller set of dimensions called **principal components**.

There isn't a particular meaning assigned to each principal component; the new components are just the two main dimensions of variations that contain most of the information in the original dataset.

```python
# Transform PCA data to a DataFrame
df_iris_pca = pd.DataFrame(
    data=iris_pca, columns=["principal component 1", "principal component 2"]
)
df_iris_pca.head()
```

	principal component 1	principal component 2
0	-2.264542	0.50574
1	-2.086426	-0.655405
2	-2.367950	-0.318477
3	-2.304197	-0.575368
4	-2.388777	0.674767

Principal Component Analysis

The principal component values bear little resemblance to the original dataset. They can be seen as a reduced representation of the original data.

The `explained_variance_ratio` attribute is used to assess the amount of information that has been preserved in the PCA dimensionality reduction.

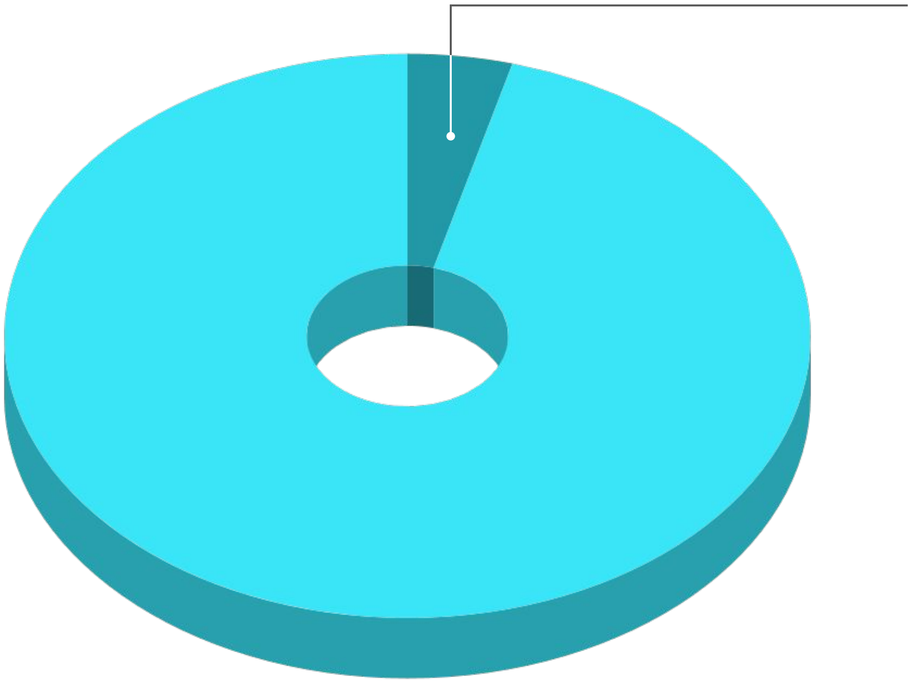
Here, the first principal component is responsible for 72.77% of the variance, and the second contains 23.03% of the variance.

```
# Fetch the explained variance
pca.explained_variance_ratio_
```

```
arry([0.33690046, 0.26230645, 0.232606390])
```

Principal Component Analysis

Together, the principal components preserve **95.80% of the information**.



In other words…

A little over 4% of the useful information was lost in the dimensionality reduction performed by PCA.

Instructor Demonstration

Speed up Machine Learning with PCA

Activity: PCA in Action

In this activity, you will use PCA to reduce the dimensions of a consumer shopping dataset.

Suggested Time:

20 minutes

Time's Up! Let's Review.

Questions?

Activity: PCA

In this activity, you will perform PCA on the Boston Marathon dataset.

Suggested Time:

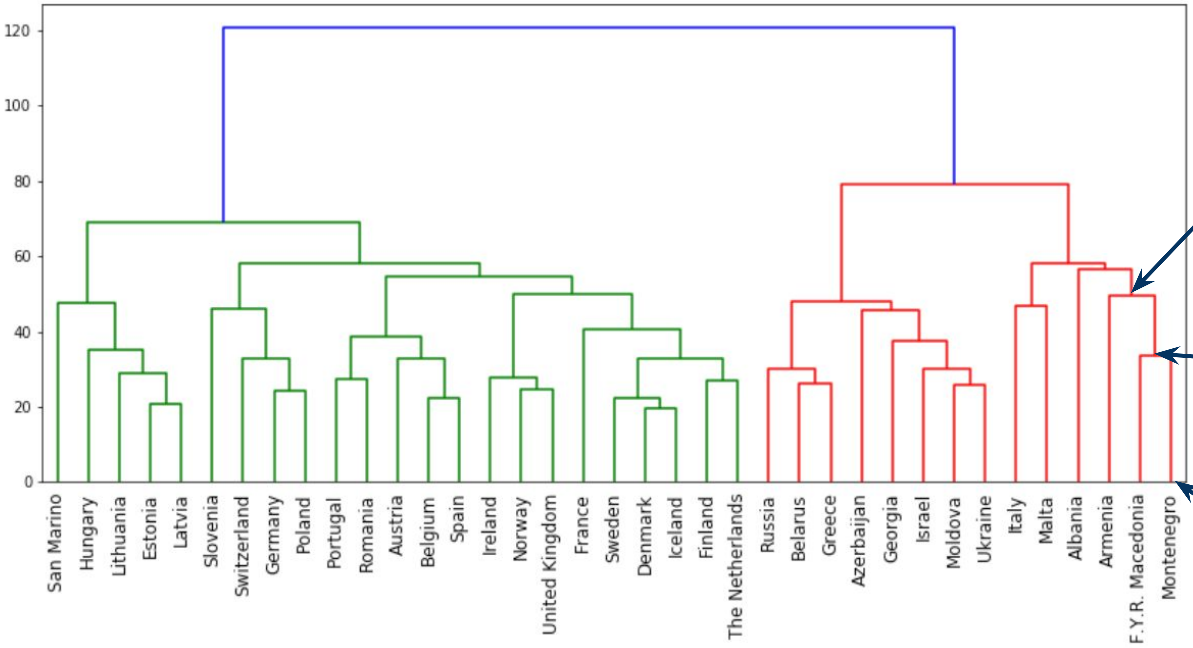
25 minutes

Time’s Up! Let’s Review.

Hierarchical Clustering

Dendrograms

This tree-like structure is called a dendrogram. Each cluster starts at the bottom.



A pair of clusters that join at a lower vertical height are closer to each other than a pair of clusters that join at a higher height.

For each cluster, the hierarchical clustering algorithm finds the closest neighbor and merges two clusters into one.

Each cluster starts at the bottom.

Dendrograms

Each country is paired with its closest neighboring cluster based on voting patterns.

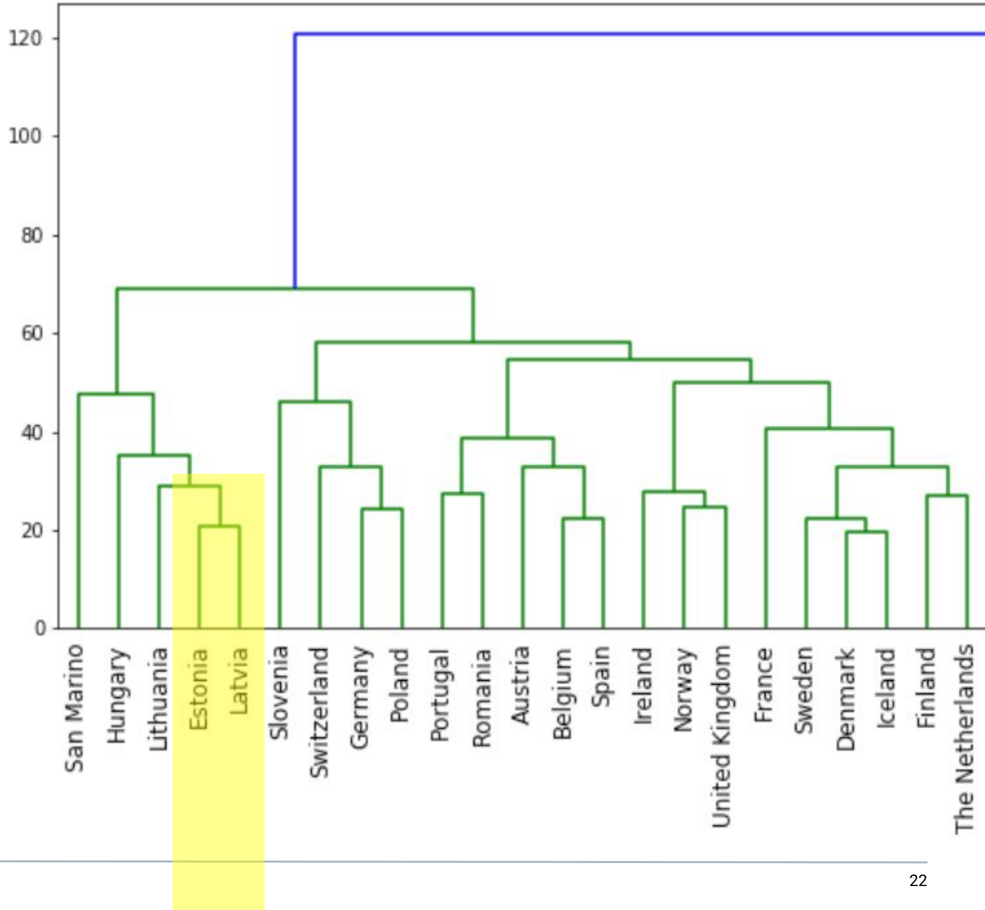
For example

Estonia's closest neighbor is Latvia, and they merge into a single cluster before merging with Lithuania.

There appear to be two large clusters:

1. The cluster on the left, in which eastern European countries are more heavily represented

2. The cluster on the right, in which eastern European countries are more heavily represented



Dendrograms

Because the mergings are based on the distances between samples, datasets should be standardized or normalized.

Here, we use scikit-learn's normalize method. The actual clustering is performed here using SciPy's linkage method.

```
normalized = normalize(df)
mergings = linkage(normalized, method='ward')
```

We're using the ward method to compute distances between clusters.

Hierarchical Clustering

To generate the dendrogram, SciPy's `dendrogram` method is used.

The first argument, `mergings`, is the linkage matrix that we just generated.

The next two arguments, `leaf_rotation` and `leaf_font_size`, refer to the text label of each sample. Here, the text is turned vertically, and its font size is set at 5.

```
dendrogram(mergings,
       leaf_rotation=90,
       leaf_font_size=5)
plt.show()
```

Hierarchical Clustering

Hierarchical clustering has different types of linkage methods:

Single

The difference between two clusters is defined by the closest distance between two clusters.

Complete

The difference between two clusters is defined by the farthest distance between two clusters.

Ward

This method is based on the squared euclidean distance between clusters. It's the method used in our example, and it is often used as a default.

Instructor Demonstration

Hierarchical Clustering

Activity: Hierarchical Customer Data

In this activity, you will use hierarchical clustering to group and plot customer data.

Suggested Time:

20 minutes

Time's Up! Let's Review.

Questions?