

AI Bootcamp

Unsupervised Learning in Practice

Module 11 Day 2



Class Objectives

By the end of class, you will be able to:

1

Segment data.

2

Prepare data for complex algorithms.

3

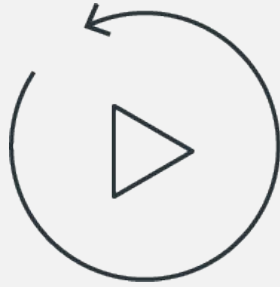
Explain the importance of preprocessing data for unsupervised learning.

4

Transform categorical variables into a numerical representation using pandas.

5

Scale data by using the **StandardScaler** module from **scikit-learn**.



Let's **recap**



Recap

In the previous lesson, you learned:

- 1 How to recognize the differences between supervised and unsupervised machine learning.
- 2 What clustering is and how to use it in data science.
- 3 How to apply the K-means algorithm to identify clusters in datasets.
- 4 How to determine the optimal number of clusters for a dataset by using the elbow method.

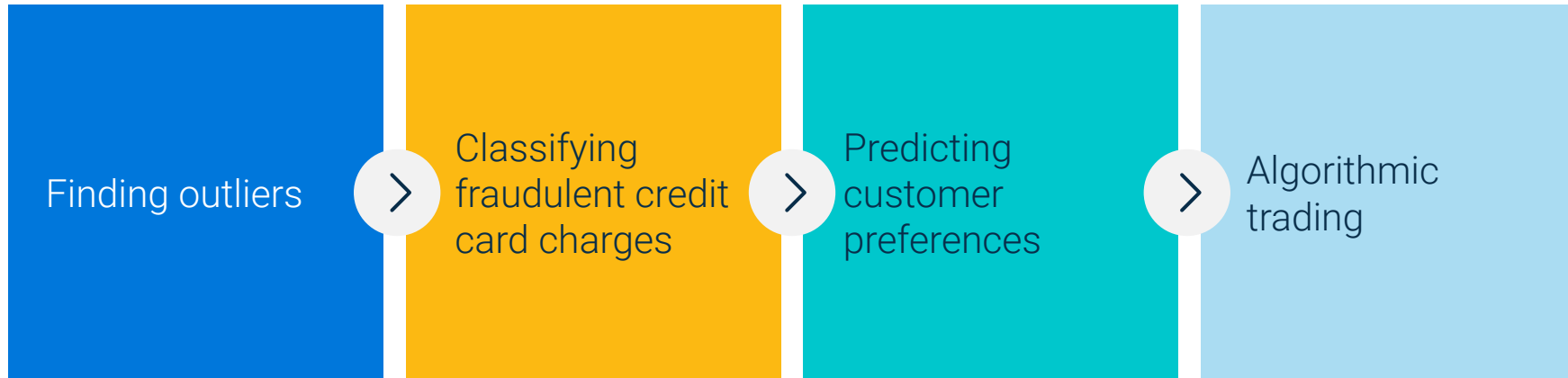


How is machine learning
used in data analytics?



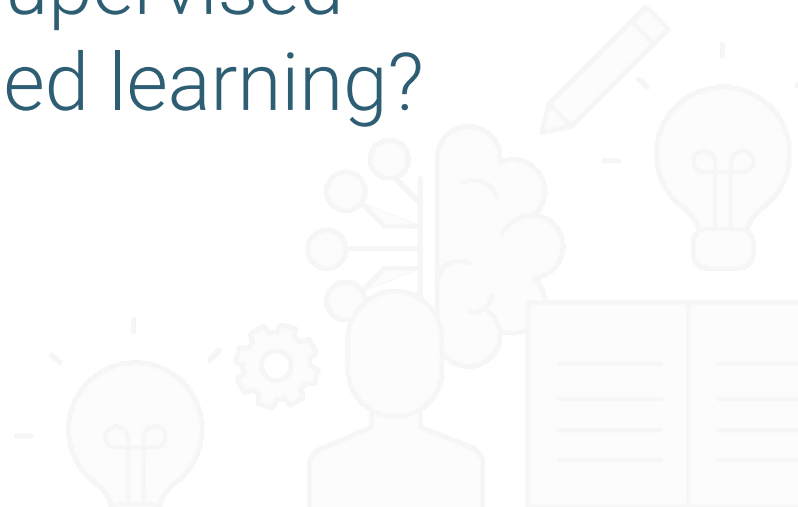
Machine Learning in Data Analytics

Examples include:





What is the difference
between unsupervised
and supervised learning?

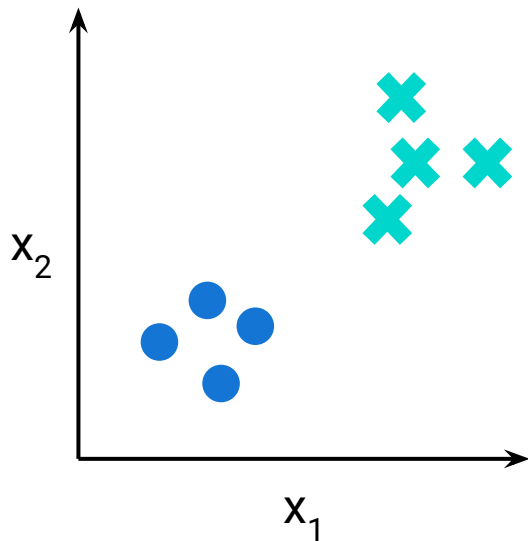


Unsupervised Learning vs. Supervised Learning

The main distinction between the two approaches is the use of labeled datasets.

Unsupervised Learning

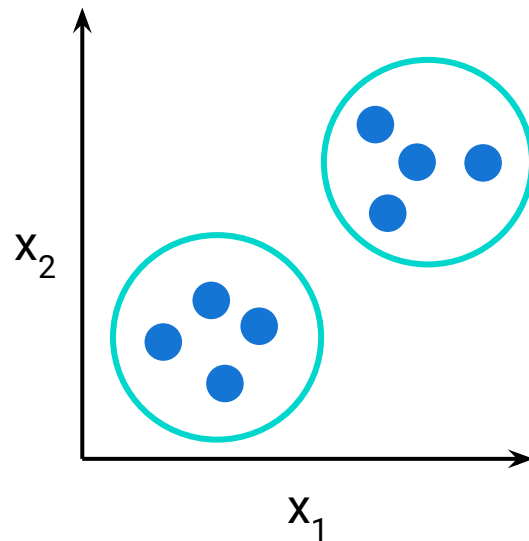
Only has labeled input data.

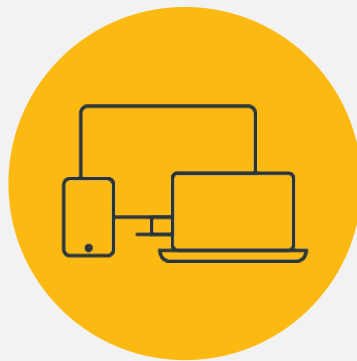


VS.

Supervised Learning

Uses labeled input and output data.





Instructor **Demonstration**

The Elbow Curve Warm-Up



Questions?





Activity:

Used Car Sales Warm-Up

In this activity, you will use the K-means algorithm to identify trends within a dataset of used car sales.

Suggested time:

15 minutes



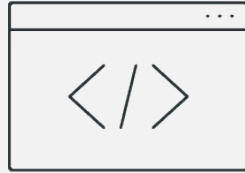


Time's up!
Let's review



Questions?

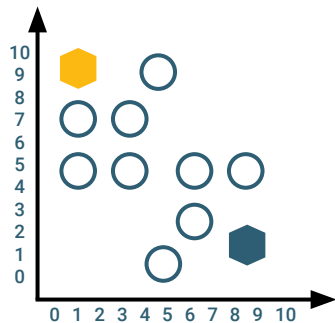




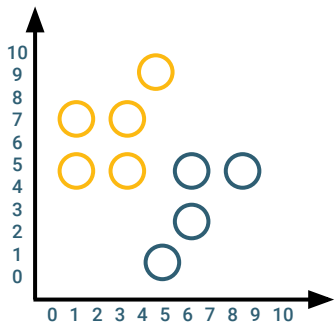
Scaling Data

Scaling Data with StandardScaler

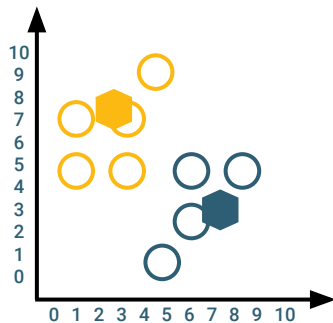
We can optimize data clustering by selecting the best value for k. The K-means algorithm is useful to group and understand data.



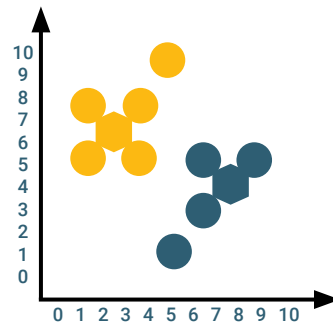
Randomly select
k clusters



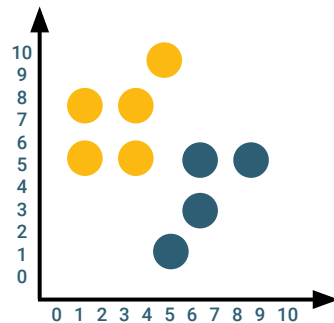
Each object assigned to
similar centroid randomly



Cluster centers updated
depending on new
cluster mean



Reassign data points and
update cluster centers



Reassign data points



We can often enhance and optimize machine learning algorithms by applying **Principal Component Analysis**, or **PCA**.

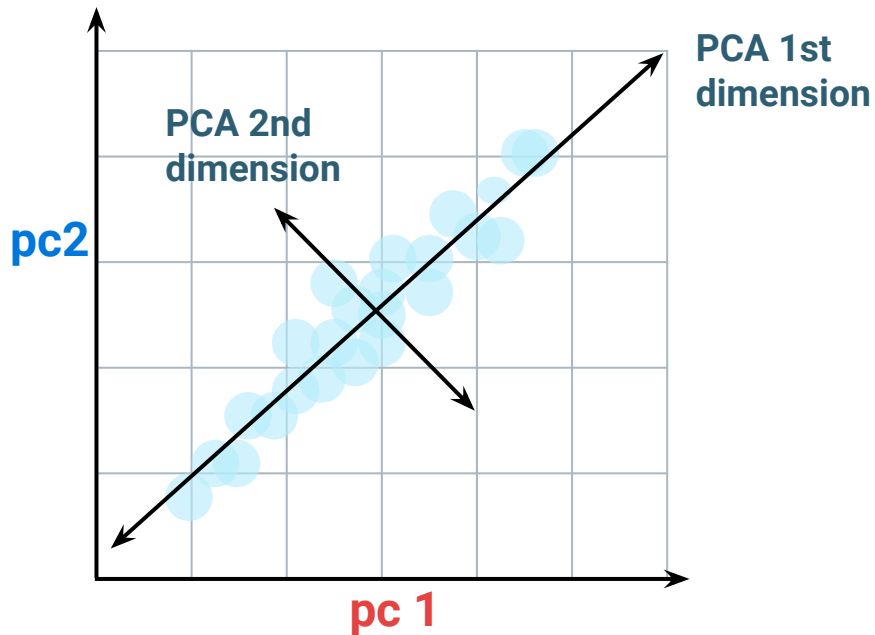


Principal Component Analysis

(PCA) is a statistical technique to streamline the machine learning process when too many factors exist in the data.

Principal Component Analysis (PCA)

PCA reduces the number of factors by transforming a large set of features into a smaller one that contains MOST of the information of the original larger dataset.



Principal Component Analysis (PCA)

PCA is a dimensionality reduction method that:



Looks at all the dimensions (or data columns) in a dataset.



Analyzes the weight of their contribution to the variance in the dataset.



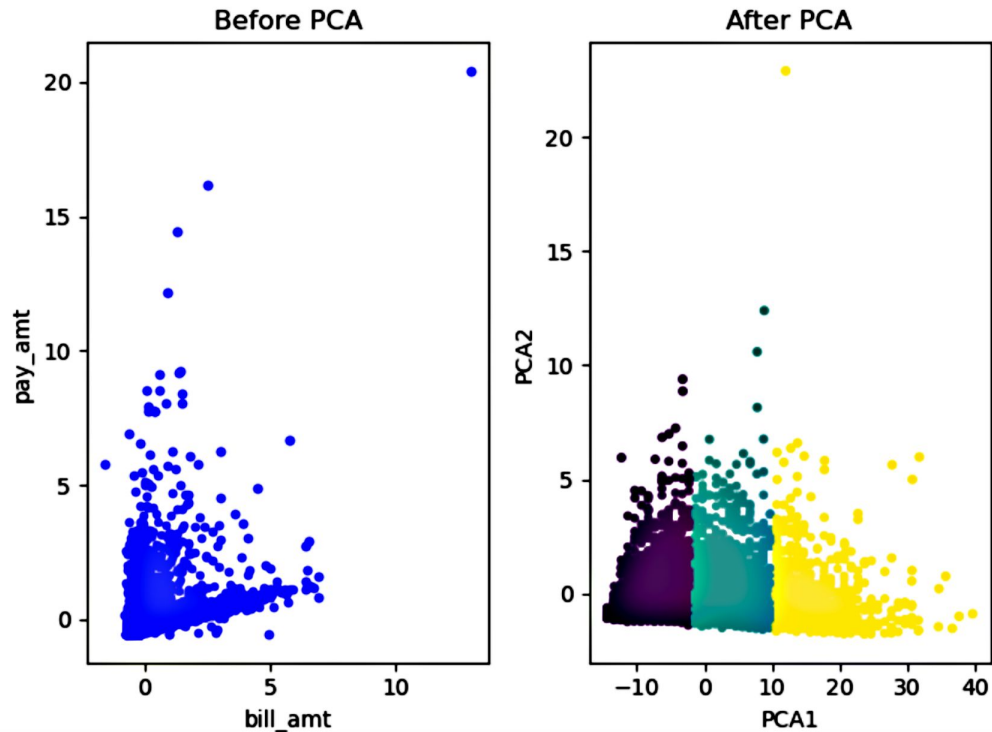
Reduces the dimensions to a smaller set that still contains as much of the information (the maximum variance) of the original dataset as possible.



PCA will NOT capture all the information from the original dataset, but it will capture as much as possible to maintain the predictive power and the meaning of the original dimensions.

Principal Component Analysis (PCA)

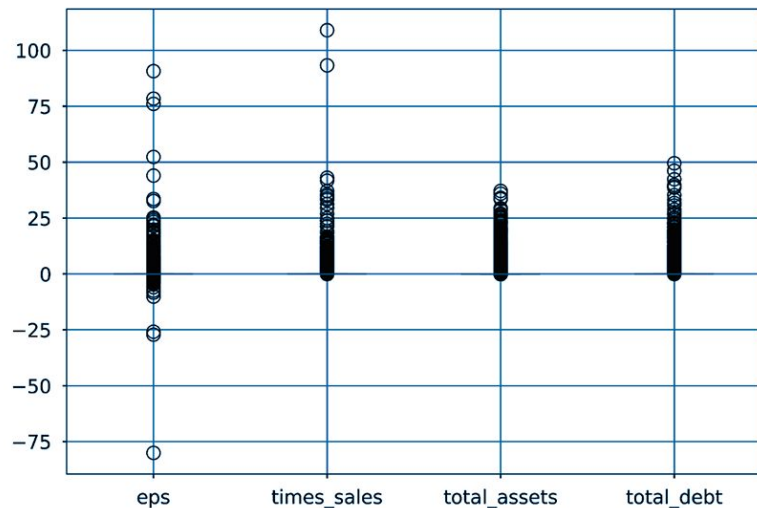
Reducing the number of factors, or **dimensional reduction**, comes at the expense of some accuracy, but the goal is to trade a little accuracy for simplicity. After applying PCA, we can get better segmentation of the data.



Standard Scaling

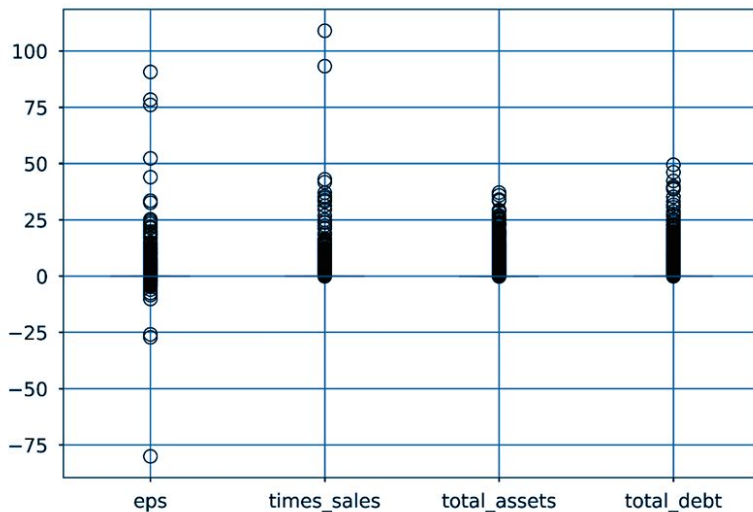
Before

Before using PCA, we'll apply standard scaling to learn how to transform the features of data.



After

After scaling, we'll combine PCA with the K-means algorithm. This will give us a strategy to better handle extremely large datasets.



Scaling Data

Remember, the K-means algorithm requires all the columns in a DataFrame to have numeric values.



We should also ensure that the numeric values have the same scale.



This prevents K-means from putting too much weight on any single variable.

Numeric Data Before Scaling

eps	times_sales	total_assets	total_debt
2.61	63.73	222822.05	46244.82
0.12	17.55	234.42	0.00
7.96	44.14	239.78	15.24
-21.25	109.27	16872.89	0.00
62.48	387.85	156035.77	41128.51

The Same Data After Scaling

eps	times_sales	total_assets	total_debt
-0.0575	-0.0797	-0.1134	-0.0864
-0.0570	0.0795	-0.1136	0.0864
-0.0594	-0.0796	-0.1136	-0.0864
-0.0567	-0.0770	0.1135	-0.0862
0.0484	0.2537	-0.0961	-0.0836

Scaling Data

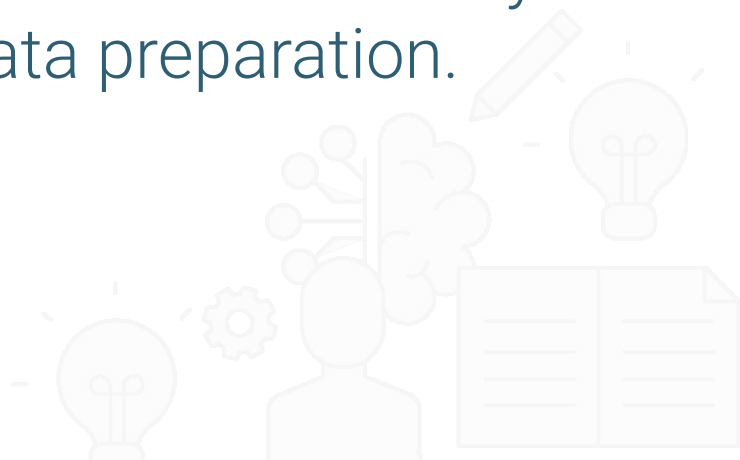
When we **scale data**, we eliminate the measurement units and adjust the numeric values to a similar scale.

We can then compare
data of **differing natures**.





Instead of manually transforming our data, we can use functions from **pandas** and the **scikit-learn** library to simplify our data preparation.



Scaling Data

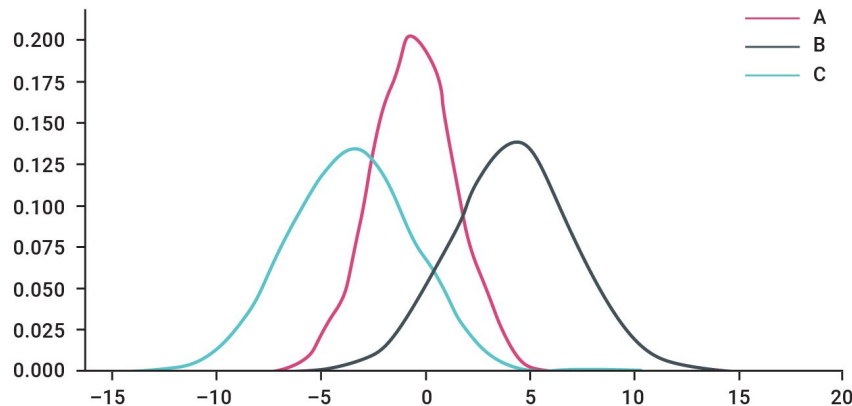
The most common way to scale data is to apply **standard scaling**, which is a method of centering values around the mean.

$$z = \frac{x - \mu}{\sigma}$$

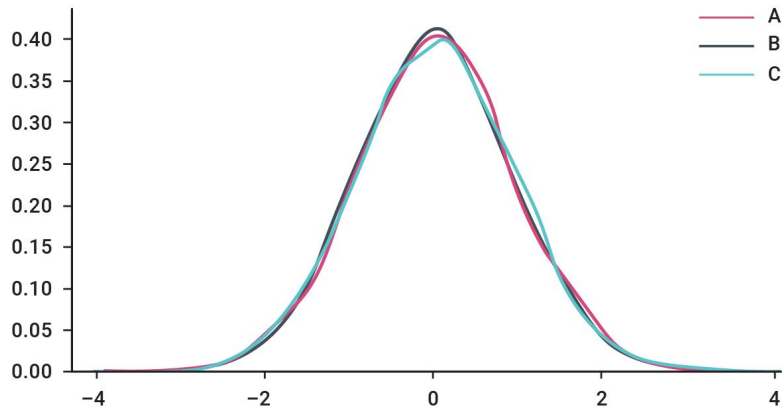
μ = Mean

σ = Standard deviation

Before scaling



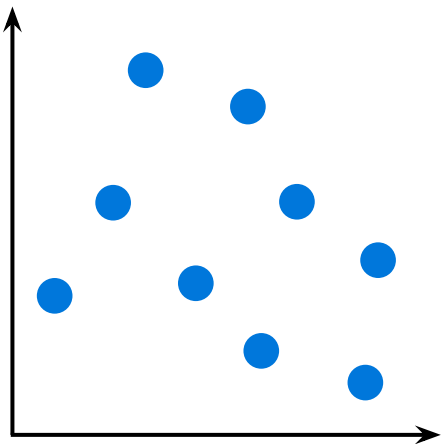
After scaling



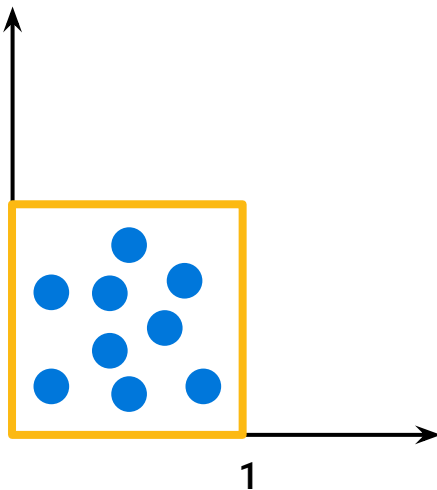
Scaling Data

Data standardization is a common practice in the data preprocessing steps that occur before training a machine learning model.

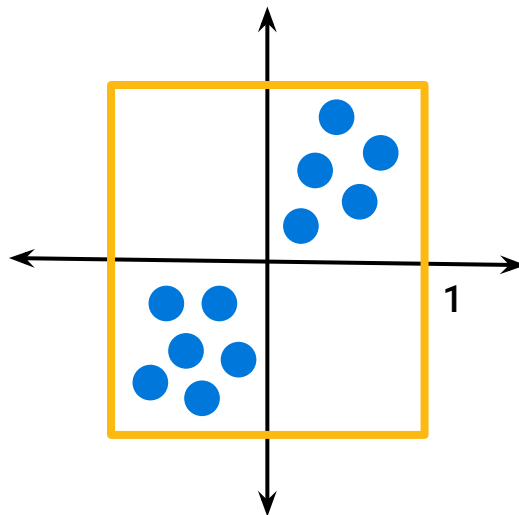
Actual data



After scaling



After standardization

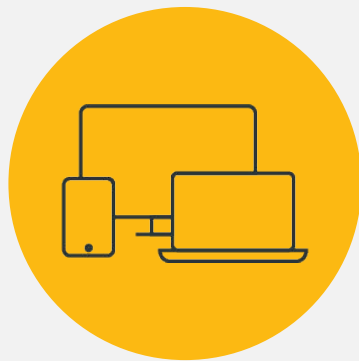




Scaling Data

Let's review one of our credit card spending datasets to illustrate how standard scaling works.

	CustomerID	Card Type	Age	Annual Income	Spending Score
0	1	Credit	19	15000	39
1	2	Credit	21	15000	81
2	3	Debit	20	16000	6
3	4	Debit	23	16000	77
4	5	Debit	31	17000	40



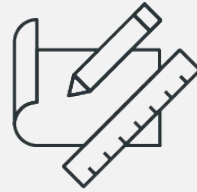
Instructor **Demonstration**

Applying Standard Scaling



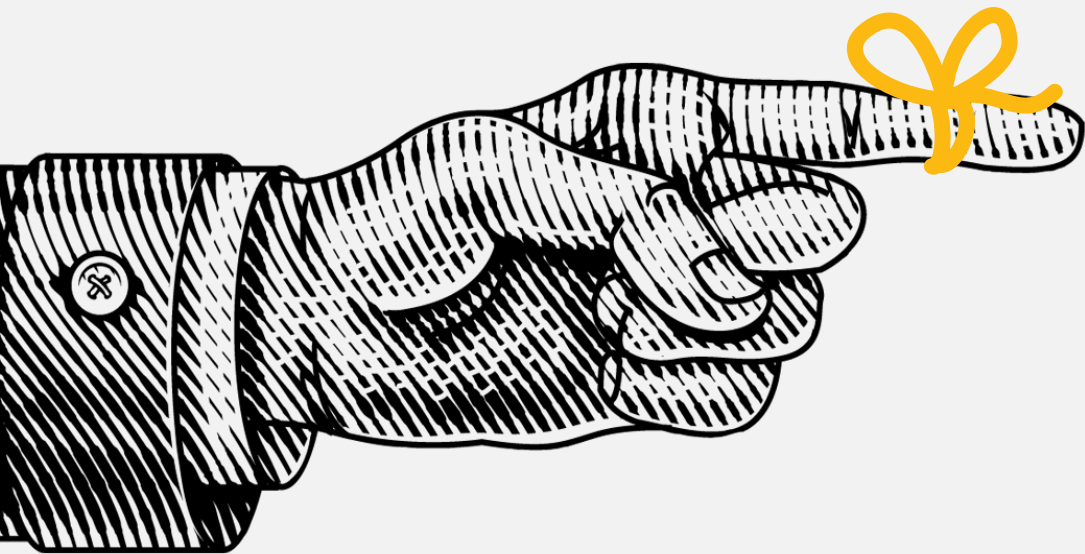
Questions?





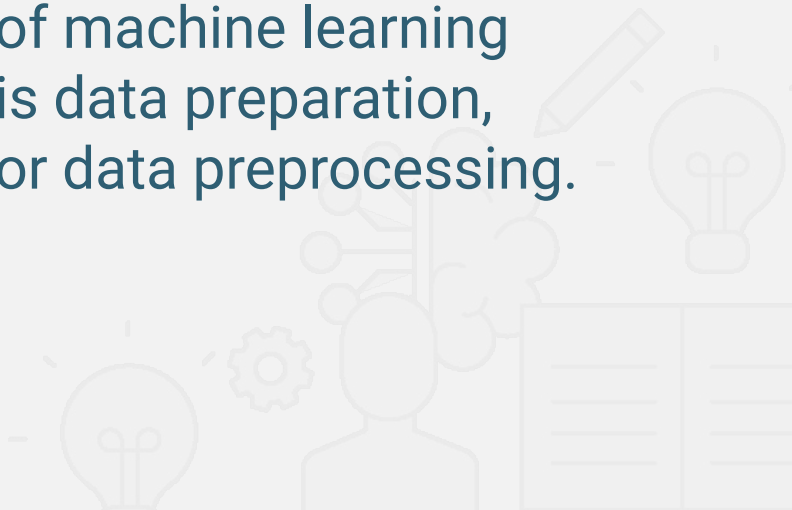
Preprocessing **Data**





Remember,

an important aspect
of machine learning
is data preparation,
or data preprocessing.



Preprocessing Data

We can import a dataset into a pandas DataFrame, but that doesn't mean all the data is ready for immediate analysis by a machine learning model.



Preprocessing Data

We should consider the following factors when feeding data to a machine learning model:

01

Most machine learning models cannot directly work with data that is in the form of strings or text.

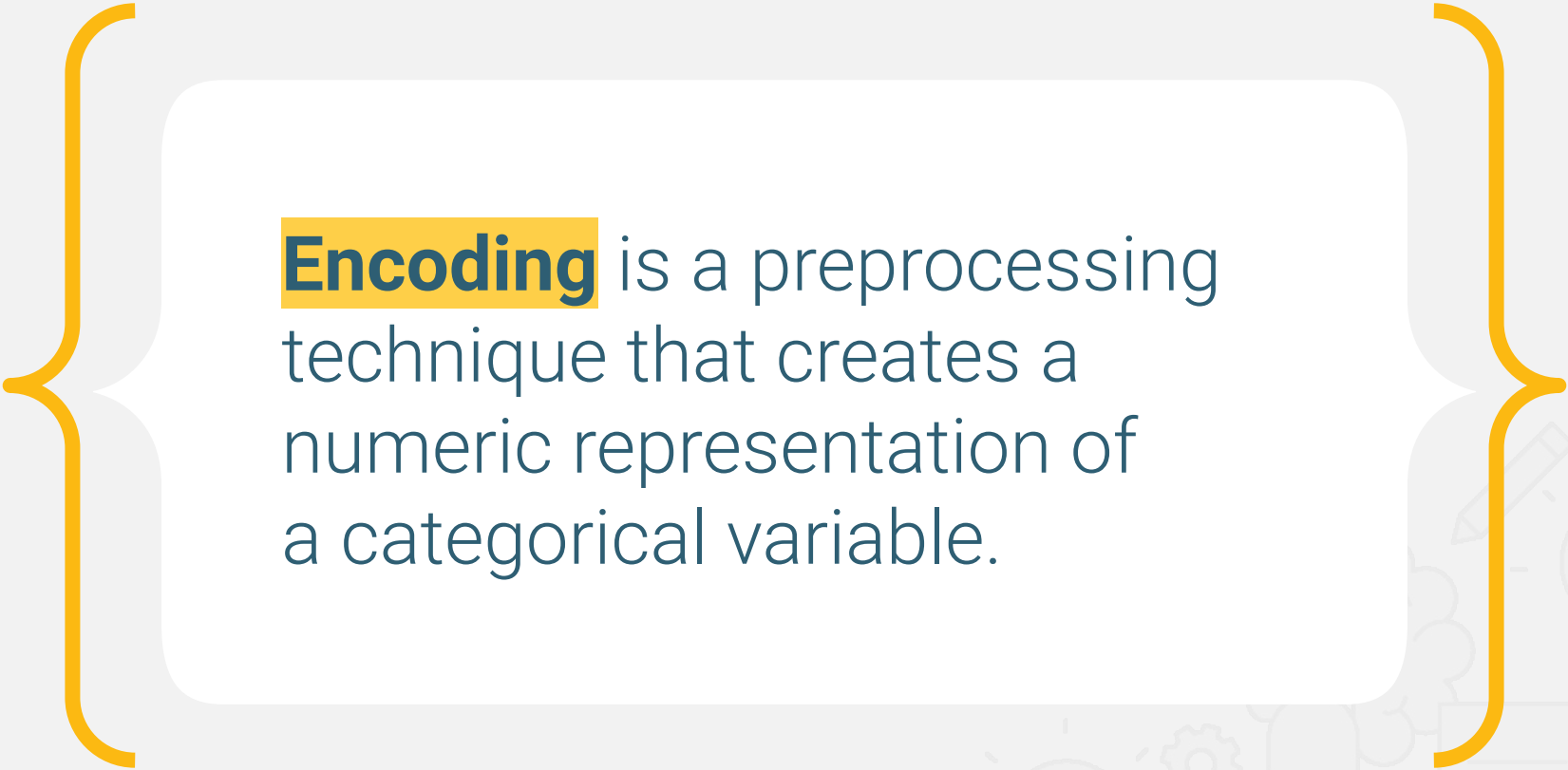
We must encode, or convert, these elements into numeric categories.

02


Machine learning algorithms have trouble learning about data with wildly different scales.

03

Missing values are difficult for machine learning models to navigate.



Encoding is a preprocessing technique that creates a numeric representation of a categorical variable.





Instructor **Demonstration**

Preprocessing Data



Questions?





Break

15 mins



Activity:

Standardizing Stock Data

In this activity, you will standardize stock data and use the K-means algorithm to cluster the data.

Suggested time:

20 minutes



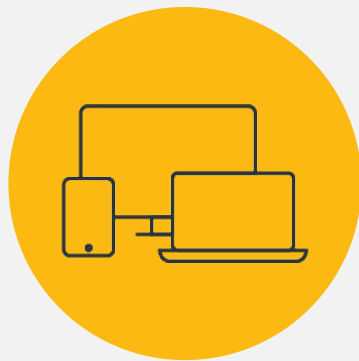


Time's up!
Let's review



Questions?





Instructor **Demonstration**

Clustering Complex Data

Clustering Complex Data

Sometimes, complex or unusual datasets might require alternative algorithms for clustering.

In this demonstration, we'll introduce two:



BIRCH



Agglomerative clustering

Clustering Complex Data: BIRCH

BIRCH stands for

Balanced **I**terative **R**educing and **C**lustering using **H**ierarchies

Clustering Complex Data: BIRCH

BIRCH is an unsupervised data mining algorithm that is similar to K-means, with a few differences.



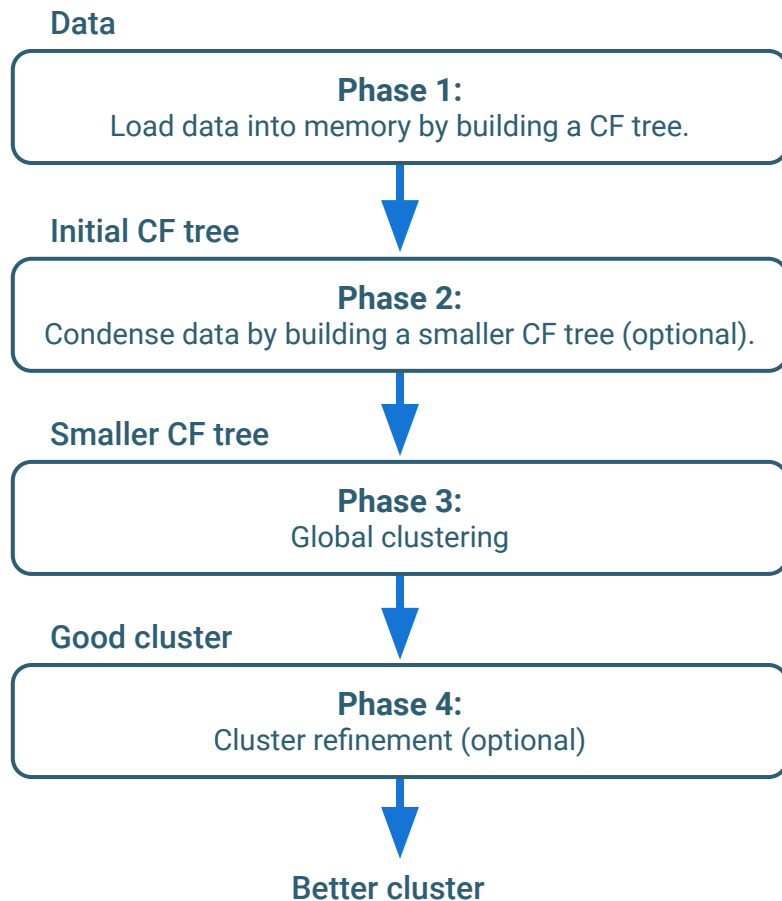
BIRCH uses hierarchical clustering.



This approach may start out with many clusters, but over the process of learning, BIRCH combines these clusters until there is only the specified number left.



The inventors of BIRCH designed it for use with extremely large datasets. This is still its main purpose because it tends to be memory-efficient.



Clustering Complex Data: Agglomerative Clustering

Agglomerative clustering is like BIRCH.



Neither one requires you to specify the appropriate cluster count, unlike K-means.



While you can specify a specific cluster count with these two approaches, they are flexible enough to divide the data into categories without much input from you.



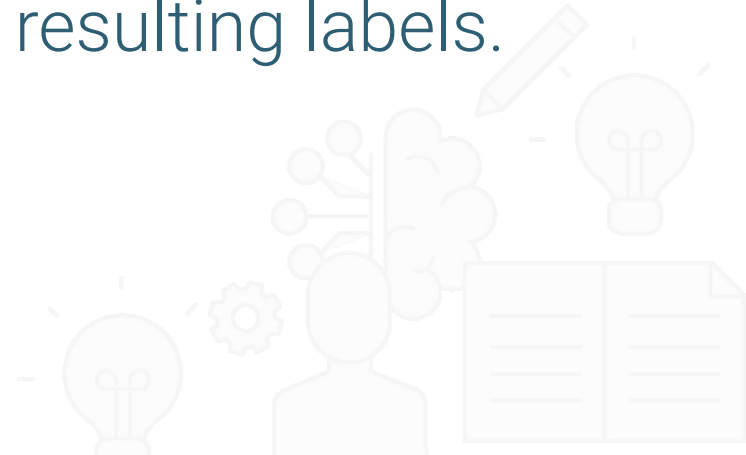
Sometimes there isn't a single, catch-all answer when deciding to use one clustering routine over another.

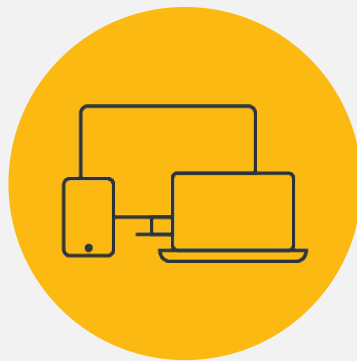


Instead, data scientists often try multiple algorithms to find out which one appears to work best on their specific data.



We'll try out **all three clustering methods** (K-means, BIRCH, agglomerative) on a single dataset and preview the resulting labels.





Instructor **Demonstration**

Compare and Contrast Alternative Clustering Algorithms



Questions?





Activity:

Segmenting Customer Data

In this activity, you will use BIRCH, agglomerative clustering, and the K-means algorithm to segment a dataset of 5,000 credit card customers.

Suggested time:

25 minutes



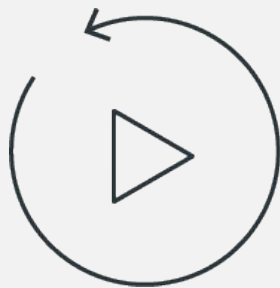


Time's up!
Let's review



Questions?





Let's **recap**



Recap

In this lesson, you learned how to:

- 1 Segment data.
- 2 Prepare data for complex algorithms.
- 3 Explain the importance of preprocessing data for unsupervised learning.
- 4 Transform categorical variables into a numerical representation using pandas.
- 5 Scale data by using the **StandardScaler** module from **scikit-learn**.



Next

In the next lesson, you will learn how to use principal component analysis (PCA) to reduce the number of features in machine learning models and improve model performance.



The End