Unsupervised Learning

(https://github.com/learn-co-curriculum/dsc-unsupervised-learning) (https://github.com/learn-co-curriculum/dsc-unsupervised-learning/issues/new)

Introduction

In this lesson, you'll get a high-level overview of unsupervised learning, an entire class of algorithms in machine learning. To date, you've only seen examples of supervised learning tasks such as regression and classification. As the name implies, unsupervised learning is a bit different than these tasks. In supervised learning, you define an `X` and `y` , and the algorithm attempts to generalize this transformation in order to predict `y` given `X` . In unsupervised learning, you do not define an `X` or `y` . Instead, you feed in a given dataset and the unsupervised learning algorithm returns some new representation of the data based on the structure and patterns within the data itself.

Objectives

You will be able to:

- Define unsupervised learning
- Compare and contrast supervised and unsupervised learning
- Identify real-world scenarios in which you would use unsupervised learning

Supervised vs. Unsupervised Learning

The main difference between supervised and unsupervised learning are their goals. Supervised learning needs concrete, ground-truth labels to train models that answer very specific questions. Unsupervised learning differs in that the task it is trying to accomplish is much less well-defined - it can usually be summed up as "are there any natural patterns in this data that are recognizable?" To illustrate this, assume that you have a basket of various different kinds of fruit. A supervised learning task would be building an apple classifier that tells us if a given fruit is or isn't an apple, based on the size, shape, color, texture, taste, and any other data that you've encoded for each piece of fruit. An unsupervised learning task on the same data would analyze only the features, and sort them into groups without being told what type of fruit each was. In general, supervised learning uses data to accomplish a clear task while unsupervised learning has no clear task, but is instead used to identify patterns.

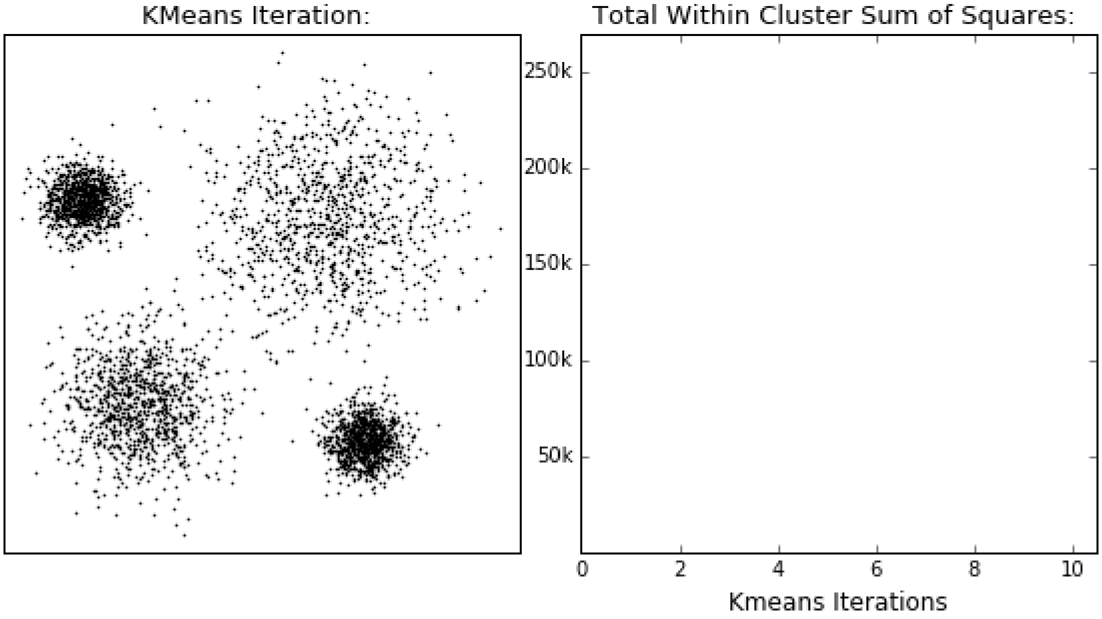
Unsupervised Learning Tasks

The two most common unsupervised learning tasks are clustering and dimensionality reduction. Clustering groups data into homogeneous groups, where members share common traits.

Dimensionality reduction attempts to reduce the overall number of features of a dataset while preserving as much information as possible. With that, let's take a deeper look into some general notes on each.

Clustering

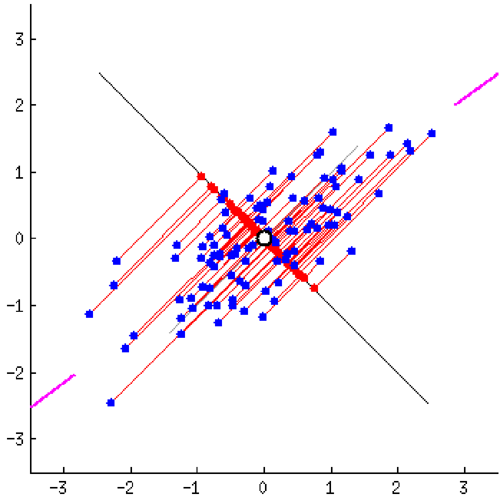
There are a few different kinds of clustering algorithms, but they all do the same thing - finding different ways to group a dataset based on patterns in the data. One common use-case for clustering is market segmentation. In market segmentation, you would try to decompose an audience into subsets for more precise targeting for business purposes, such as advertising. Even though there's no way to verify that these groups are correct, in practice it usually does quite well, often providing useful subgroups which can then be individually examined.



> Source: [GIF by David Sheehan (https://dashee87.github.io/data%20science/general/Clustering-with-Scikit-with-GIFs/)](https://dashee87.github.io/data%20science/general/Clustering-with-Scikit-with-GIFs/)

Dimensionality Reduction

The most common dimensionality reduction algorithm is Principal Component Analysis (PCA). Dimensionality reduction algorithms work by projecting data from its current n-dimensional subspace into a smaller subspace, while losing as little information as possible in the process. Dimensionality reduction algorithms still lose *some* information, but you can quantify this information loss to make an informed decision about the number of dimensions reduced versus the overall information lost. Dimensionality reduction algorithms are a must-have in any data scientist's toolbox, because they provide a way for us to deal with the **Curse of Dimensionality**. The curse of dimensionality is a key concept as datasets scale. In short, as the number of features in a dataset increases, the processing power and search space required to optimize a given machine learning algorithm explodes exponentially. Because this often creates intractable computational problems, dimensionality reduction techniques such as PCA can be an essential preprocessing technique.



> Source: **[GIF by amoeba](https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues/140579#140579) (https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues/140579#140579)**

Summary

In this lesson, you explored the differences between supervised and unsupervised learning. You also learned about the types of problems we can solve with unsupervised learning, including clustering and dimensionality reduction.