Unsupervised Learning Algorithms

Unsupervised algorithms are relevant when we don't have an outcome or labeled variable we are trying to predict.

They are helpful to find structures within our data set and when we want to partition our data set into smaller pieces.

Types of Unsupervised Learning:

Type of Unsupervised Learning	**Data**	**Example**	**Algorithms**
Clustering	Use unlabeled data, Identify unknown structures in data	Segmenting customers into different groups	K-means, Hierarchical Agglomerative Clustering, DBSCAN, Mean shift
Dimensionality Reduction	Use structural characteristics to simplify data	Reducing size without losing too much information from our original data set	Principal Components Analysis, Non-negative Matrix, Factorization

Dimensionality reduction is important in the context of large amounts of data.

The Curse of Dimensionality

In theory, a large number of features should improve performance. As models have more data to learn from, they should be more successful. But in practice, too many features lead to worse performance. There are several reasons why too many features end up leading to worse performance. If you have too many features, several things can be wrong, for example:

- Some features can be spurious correlations, which means they correlate into the data set but not outside your data set, as long as new data comes in.

- Too many features create more noise than signal.

- Algorithms find it hard to sort through non-meaningful features if you have too many features.

- The number of training examples required increases exponentially with dimensionality.

- Higher dimensions slows performance.

- Larger data sets are computationally more expensive.

- Higher incidence of outliers.

To fix these problems in real life, it's best to reduce the dimension of the data set.

Similar to feature selection, you can use Unsupervised Machine Learning models such as Principal Components Analysis.

Common uses of clustering cases in the real world

1. Anomaly detection

Example: Fraudulent transactions.

Suspicious fraud patterns such as small clusters of credit card transactions with high volume of attempts, small amounts, for new merchants. This creates a new cluster and this is presented as an anomaly so perhaps there's fraudulent transactions happening.

2. Customer segmentation

You could segment the customers by recency, frequency, and average amount of visits in the last 3 months. Another common type of segmentation is by demographic and the level of engagement, for example, single costumers, new parents, empty nesters, etc. And the combinations of each with the preferred marketing channel, so you can use these insights for future marketing campaigns.

3. Improve supervised learning

You can perform a Logistic regression for each cluster. This means training one model for each segment of your data to try to improve the classification.

Common uses of Dimension Reduction in the real world

1. Turn high-resolution images into compressed images

This means to come to a reduced, more compact version of those images, so they can still contain most of the data that can tell us what the image is about.

2. Image tracking

Reduce the noise to the primary factors that are relevant in a video capture. The benefits of reducing the data set can greatly speed up the computational efficiency of the detection algorithms.

K-means Clustering

K-means clustering is an iterative process in which similar observations are grouped together. To do that, this algorithm starts by taking 2 random points known as centroids, and starts calculating the distance of each observation to the centroid, and assigning each cluster to the nearest centroid. After the first iteration, every point belongs to a cluster.

Next, the number of centroids increases by one, and the centroid for each cluster is recalculated as the points with the average distance to all points in a given cluster. Then, we keep repeating this process until no example is assigned to another cluster.

And this process is repeated k-times, hence the name k-means. This algorithm converges when clusters do not move anymore.

We can also create multiple clusters, and we can have multiple solutions. By multiple solutions, we mean that the clusters are not going to move anymore (they converged), but we can converge in different places, where we no longer move those centroids.

Advantages and Disadvantages of K-Means

The main advantage of k-means algorithm is that it is easy to compute. One disadvantage is that this algorithm is sensitive to the choice of the initial points, so different initial configurations may yield different results.

To overcome this, there is a smarter initialization of K-mean clusters called K-means ++, which helps to avoid getting stuck at local optima. This is the default implementation of the K-means.

Model Selection, choosing *K* number of clusters

Sometimes you want to split your data into a predetermined number of groups or segments. Often, the number of clusters (K) is unclear, and you need an approach to select it.

A common metric is ***Inertia***, defined as the sum of squares distance from each point to its cluster centroid.

Smaller values of Inertia correspond to tighter clusters, this means that we are penalizing spread out clusters and rewarding clusters that are tighter to their centroids.

The drawback of this metric is that its value sensitive to number of points in clusters. The more points you add, the more you will continue penalizing the inertia of a cluster, even if those points that are relatively closer to the centroids than the existing points.

Another metric is ***Distortion,*** defined as the average of squared distance from each point to its cluster.

Smaller values of distortion corresponds to tighter clusters.

An advantage of distortion is that it doesn't generally increase, as more points are added (relative to inertia). This means that it doesn't increase distortion, as closer points will actually decrease the average distance to the cluster centroid.

Inertia Vs. Distortion

Both Inertia and Distortion are measures of entropy per cluster.

Inertia will always increase, as more members are added to each cluster, while this will not be the case with distortion.

When the similarity of the points in the cluster are very relevant, you should use distortion and if you are more concerned that clusters should have a similar number of points, then you should use inertia.

Finding the right cluster

To find the cluster with a low entropy metric, you can run a few k-means clustering models with different initial configurations, compare the results, and determine which one of the different initializations of configurations leads to the lowest inertia or distortion.