

Unsupervised Learning with Python



Vihar Kurama

Follow

May 12, 2018 · 7 min read

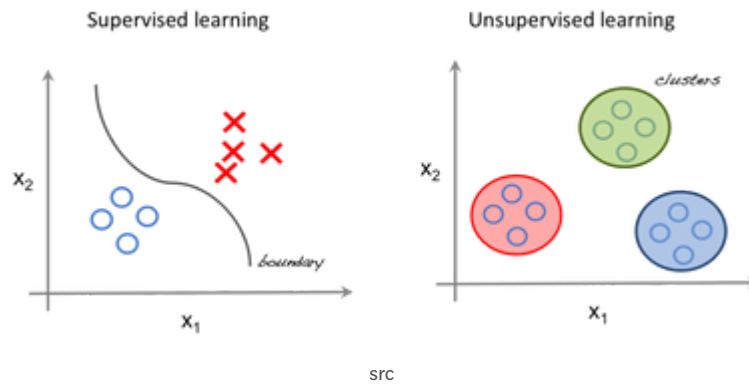
Unsupervised Learning is a class of Machine Learning techniques to find the patterns in data. The data given to unsupervised algorithm are not labelled, which means only the input variables(X) are given with no corresponding output variables. In unsupervised learning, the algorithms are left to themselves to discover interesting structures in the data.



Yan Lecun, director of AI research, explains that unsupervised learning—teaching machines to learn for themselves without having to be explicitly told if everything they do is right or wrong—is the key to “true” AI.

Supervised Vs Unsupervised Learning.

In supervised learning, the system tries to learn from the previous examples that are given. (On the other hand, in unsupervised learning, the system attempts to find the patterns directly from the example given.) So if the dataset is labelled it comes under a supervised problem, if the dataset is unlabelled then it is an unsupervised problem.



The image to the left is an example of supervised learning; we use regression techniques to find the best fit line between the features. While in unsupervised learning the inputs are segregated based on features and the prediction is based on which cluster it belonged.

Important Terminology

Feature: An input variable used in making predictions.

Predictions: A model's output when provided with an input example.

Example: One row of a data set. An example contains one or more features and possibly a label.

Label: Result of the feature.

Preparing data for Unsupervised Learning

In this article we use, Iris dataset for making our very first predictions. The dataset contains a set of 150 records under 5 attributes—Petal Length , Petal Width , Sepal Length , Sepal width and Class. Iris Setosa, Iris Virginica and Iris Versicolor are the three classes. For our Unsupervised Algorithm we give these four features of the Iris flower and predict which class it belongs to.

We use sklearn Library in Python to load Iris dataset, and matplotlib for data visualisation. Below is the code snippet for exploring the dataset.

```

1  # Importing Modules
2  from sklearn import datasets
3  import matplotlib.pyplot as plt
4
5  # Loading dataset
6  iris_df = datasets.load_iris()
7
8  # Available methods on dataset
9  print(dir(iris_df))
10
11 # Features
12 print(iris_df.feature_names)
13
14 # Targets
15 print(iris_df.target)
16
17 # Target Names
18 print(iris_df.target_names)

```

```

['DESCR', 'data', 'feature_names', 'target', 'target_names']
['sepal length (cm)', 'sepal width (cm)', 'petal length
(cm)', 'petal width (cm)']

```

```

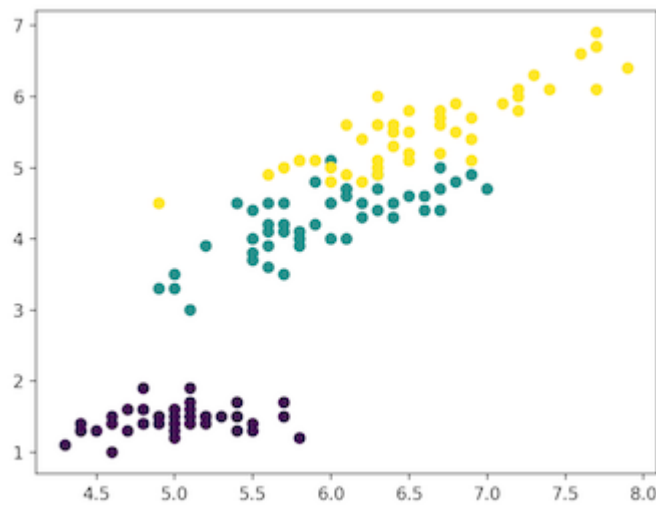
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2]

```

```

['setosa' 'versicolor' 'virginica']

```

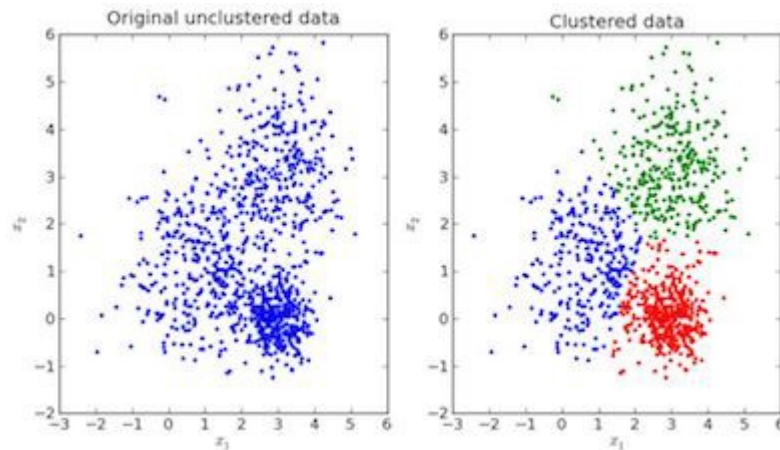


Violet: Setosa, Green: Versicolor, Yellow: Virginica

Clustering

In clustering, the data is divided into several groups. In plain words, the aim is to segregate groups with similar traits and assign them into clusters.

Visual Example,



In the above image, the image to the left is raw data where the classification isn't done, the image in the right is clustered (the data is classified based on its features). When an input is given which is to be predicted then it checks in the cluster it belongs based on its features, and the prediction is made.

K-Means Clustering in Python

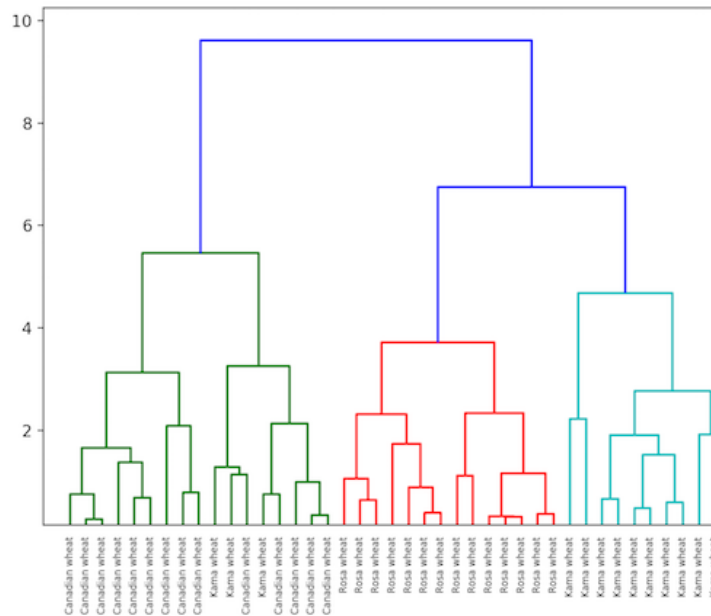
K means is an iterative clustering algorithm that aims to find local maxima in each iteration. Initially desired number of clusters are chosen. Since we know that there are 3 classes involved, we program

Hierarchical clustering, as the name implies is an algorithm that builds a hierarchy of clusters. This algorithm begins with all the data assigned to a cluster of their own. Then two closest clusters are joined into the same cluster. In the end, this algorithm ends when there is only a single cluster left.

The completion of hierarchical clustering can be shown using dendrogram. Now let's see an example of hierarchical clustering of grain data. The dataset can be found [here](#).

Hierarchical Clustering Implementation in Python.

```
1  # Importing Modules
2  from scipy.cluster.hierarchy import linkage, dendrogram
3  import matplotlib.pyplot as plt
4  import pandas as pd
5
6  # Reading the DataFrame
7  seeds_df = pd.read_csv(
8      "https://raw.githubusercontent.com/vihar/unsupervised-learning/master/datasets/seeds.csv"
9  )
10 # Remove the grain species from the DataFrame, save for later
11 varieties = list(seeds_df.pop('grain_variety'))
12
13 # Extract the measurements as a NumPy array
14 samples = seeds_df.values
15
16 """
17 Perform hierarchical clustering on samples using the
18 linkage() function with the method='complete' keyword
19 Assign the result to mergings.
20 """
21 mergings = linkage(samples, method='complete')
22
23 ..
```



Difference between K Means and Hierarchical clustering

- Hierarchical clustering can't handle big data well but K Means clustering can. This is because the time complexity of K Means is linear i.e. $O(n)$ while that of hierarchical clustering is quadratic i.e. $O(n^2)$.
- In K Means clustering, as we start with an arbitrary choice of clusters, the results generated by running the algorithm multiple times might differ. While results are reproducible in Hierarchical clustering.
- K Means is found to work well when the shape of the clusters is hyper spherical (like circle in 2D, sphere in 3D).
- K-Means doesn't allow noisy data, while in Hierarchical we can directly use noisy dataset for clustering.

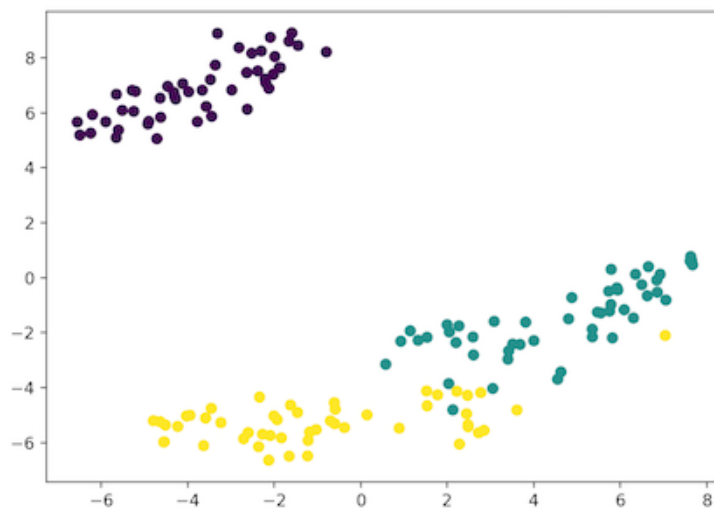
t-SNE Clustering

It is one of the unsupervised learning method for visualisation. t-SNE stands for **t-distributed stochastic neighbor embedding**. It maps high dimensional space into a 2 or 3 dimensional space which can be visualised. Specifically, it models each high-dimensional object by a two- or three-dimensional point in such a way that similar objects are

modeled by nearby points and dissimilar objects are modeled by distant points with high probability.

t-SNE Clustering Implementation in Python for Iris Dataset.

```
1  # Importing Modules
2  from sklearn import datasets
3  from sklearn.manifold import TSNE
4  import matplotlib.pyplot as plt
5
6  # Loading dataset
7  iris_df = datasets.load_iris()
8
9  # Defining Model
10 model = TSNE(learning_rate=100)
11
12 # Fitting Model
13 transformed = model.fit_transform(iris_df.data)
14
```



Violet: Setosa, Green: Versicolor, Yellow: Virginica

Here as the Iris dataset has four features(4d) it is transformed and represented in two dimensional figure. Similarly t-SNE model can be applied to a dataset which has n-features.

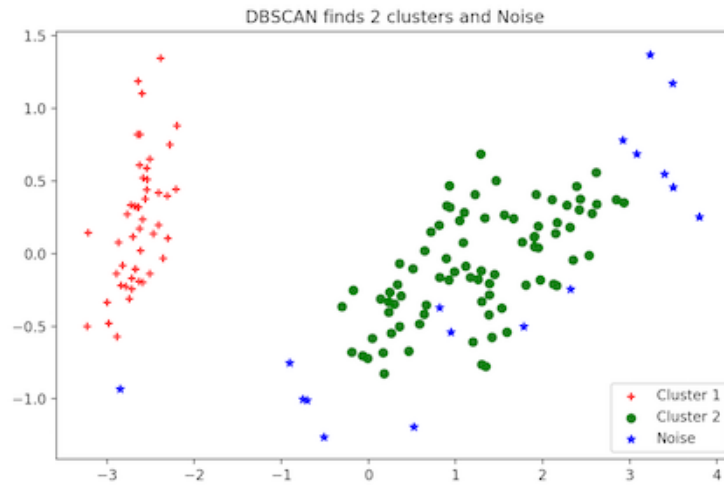
DBSCAN Clustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a popular clustering algorithm used as an replacement to K-means in predictive analytics. It doesn't require that you input the number of clusters in order to run. But in exchange, you have to tune two other parameters.

The scikit-learn implementation provides a default for the `eps` and `min_samples` parameters, but you're generally expected to tune those. The `eps` parameter is the maximum distance between two data points to be considered in the same neighborhood. The `min_samples` parameter is the minimum amount of data points in a neighborhood to be considered a cluster.

DBSCAN Clustering in Python

```
1  # Importing Modules
2  from sklearn.datasets import load_iris
3  import matplotlib.pyplot as plt
4  from sklearn.cluster import DBSCAN
5  from sklearn.decomposition import PCA
6
7  # Load Dataset
8  iris = load_iris()
9
10 # Declaring Model
11 dbscan = DBSCAN()
12
13 # Fitting
14 dbscan.fit(iris.data)
15
16 # Transoring Using PCA
17 pca = PCA(n_components=2).fit(iris.data)
18 pca_2d = pca.transform(iris.data)
19
20 # Plot based on Class
21 for i in range(0, pca_2d.shape[0]):
```



More Unsupervised Techniques:

- Principal Component Analysis (PCA)
- Anomaly detection
- Autoencoders
- Deep Belief Nets
- Hebbian Learning
- Generative Adversarial Networks(GANs)
- Self-Organizing maps

. . .

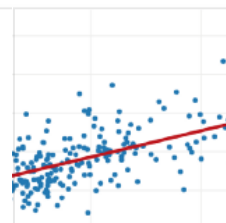
Important Links:

Supervised Learning In Python.

Supervised Learning with Python

Why Artificial Intelligence and Machine Learning ?

towardsdatascience.com

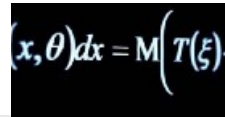


Introduction To Machine Learning

Machine Learning is an idea to learn from

$$\xi_l = \frac{(\xi_l - a)}{\sigma^2} f_a$$

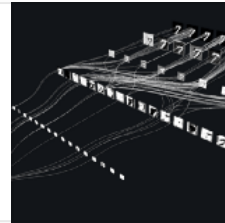
examples and experience, without being explicitl...
towardsdatascience.com


$$\int (x, \theta) dx = M \left(T(\xi) \right).$$

Deep Learning with Python

The human brain imitation.

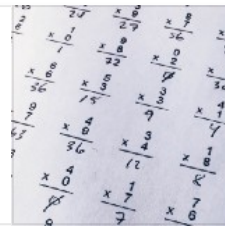
towardsdatascience.com



Linear Algebra for Deep Learning

The Math behind every deep learning program.

towardsdatascience.com



Closing Notes

Thanks for reading. If you found this story helpful, please click the below 🙌 to spread the love.

