1. Please describe the central limit theorem and provide an example.

In probability theory, the central limit theorem (CLT) establishes that, in some situations, when independent random variables are added, their properly normalized sum tends toward a normal distribution (informally a "bell curve") even if the original variables themselves are not normally distributed. The theorem is a key concept in probability theory because it implies that probabilistic and statistical methods that work for normal distributions can be applicable to many problems involving other types of distributions.

For example, suppose that a sample is obtained containing many observations, each observation being randomly generated in a way that does not depend on the values of the other observations, and that the arithmetic mean of the observed values is computed. If this procedure is performed many times, the central limit theorem says that the distribution of the average will be closely approximated by a normal distribution. A simple example of this is that if one flips a coin many times the probability of getting a given number of heads in a series of flips will approach a normal curve, with mean equal to half the total number of flips in each series. (In the limit of an infinite number of flips, it will equal a normal curve.)

The central limit theorem has a number of variants. In its common form, the random variables must be identically distributed. In variants, convergence of the mean to the normal distribution also occurs for non-identical distributions or for non-independent observations, given that they comply with certain conditions.

The earliest version of this theorem, that the normal distribution may be used as an approximation to the binomial distribution, is now known as the de Moivre–Laplace theorem.

In more general usage, a central limit theorem is any of a set of weak-convergence theorems in probability theory. They all express the fact that a sum of many independent and identically distributed (i.i.d.) random variables, or alternatively, random variables with specific types of dependence, will tend to be distributed according to one of a small set of attractor distributions. When the variance of the i.i.d. variables is finite, the attractor distribution is the normal distribution. In contrast, the sum of a number of i.i.d. random variables with power law tail distributions decreasing as |x|−α − 1 where 0 < α < 2 (and therefore having infinite variance) will tend to an alpha-stable distribution with stability parameter (or index of stability) of α as the number of variables grows.[1]

1. Describe a classification algorithm that you have previously put into production and why it was chosen.

<https://stackabuse.com/overview-of-classification-methods-in-python-with-scikit-learn/>

[Scikit-Learn](https://scikit-learn.org/stable/) is a library for Python that was first developed by David Cournapeau in 2007. It contains a range of useful algorithms that can easily be implemented and tweaked for the purposes of classification and other machine learning tasks.

Scikit-Learn uses [SciPy](https://www.scipy.org/" \t "_blank) as a foundation, so this base stack of libraries must be installed before Scikit-Learn can be utilized.

### Defining our Terms

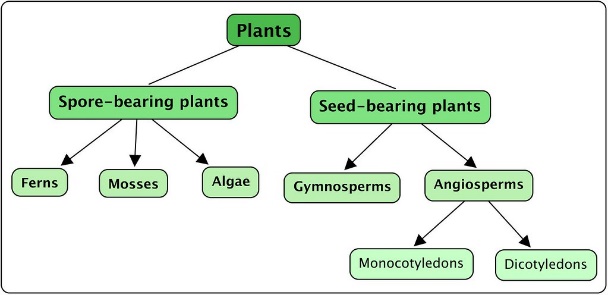
Before we go any further into our exploration of Scikit-Learn, let's take a minute to define our terms. It is important to have an understanding of the vocabulary that will be used when describing Scikit-Learn's functions.

To begin with, a machine learning system or network takes inputs and outputs. The inputs into the machine learning framework are often referred to as ["features"](https://en.wikipedia.org/wiki/Feature_(machine_learning)) .

Features are essentially the same as variables in a scientific experiment, they are characteristics of the phenomenon under observation that can be quantified or measured in some fashion.

When these features are fed into a machine learning framework the network tries to discern relevant patterns between the features. These patterns are then used to generate the outputs of the framework/network.

The outputs of the framework are often called "labels", as the output features have some label given to them by the network, some assumption about what category the output falls into.

Credit: [Siyavula Education](https://www.flickr.com/photos/121935927@N06" \t "_blank)

In a machine learning context, classification is a type of [supervised learning](https://dataconomy.com/2015/01/whats-the-difference-between-supervised-and-unsupervised-learning/). Supervised learning means that the data fed to the network is already labeled, with the important features/attributes already separated into distinct categories beforehand.

This means that the network knows which parts of the input are important, and there is also a target or ground truth that the network can check itself against. An example of classification is sorting a bunch of different plants into different categories like ferns or angiosperms. That task could be accomplished with a Decision Tree, a type of classifier in Scikit-Learn.

In contrast, unsupervised learning is where the data fed to the network is unlabeled and the network must try to learn for itself what features are most important. As mentioned, classification is a type of supervised learning, and therefore we won't be covering unsupervised learning methods in this article.

The process of training a model is the process of feeding data into a neural network and letting it learn the patterns of the data. The training process takes in the data and pulls out the features of the dataset. During the training process for a supervised classification task the network is passed both the features and the labels of the training data. However, during testing, the network is only fed features.

The testing process is where the patterns that the network has learned are tested. The features are given to the network, and the network must predict the labels. The data for the network is divided into training and testing sets, two different sets of inputs. You do not test the classifier on the same dataset you train it on, as the model has already learned the patterns of this set of data and it would be extreme bias.

Instead, the dataset is split up into training and testing sets, a set the classifier trains on and a set the classifier has never seen before.

### Different Types of Classifiers

Credit: [CreativeMagic](https://pixabay.com/users/creativemagic-480360/" \t "_blank)

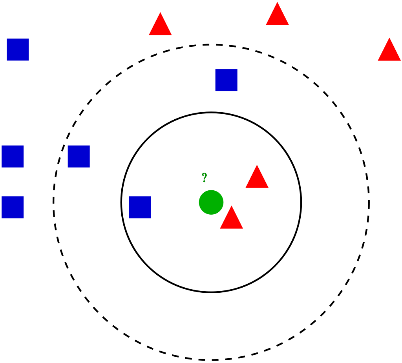
Scikit-Learn provides easy access to numerous different classification algorithms. Among these classifiers are:

* [K-Nearest Neighbors](https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn/)
* [Support Vector Machines](https://stackabuse.com/implementing-svm-and-kernel-svm-with-pythons-scikit-learn/)
* [Decision Tree Classifiers](https://stackabuse.com/decision-trees-in-python-with-scikit-learn/)/[Random Forests](https://stackabuse.com/random-forest-algorithm-with-python-and-scikit-learn/)
* [Naive Bayes](https://stackabuse.com/the-naive-bayes-algorithm-in-python-with-scikit-learn/)
* [Linear Discriminant Analysis](https://stackabuse.com/implementing-lda-in-python-with-scikit-learn/)
* Logistic Regression

There is a lot of literature on how these various classifiers work, and brief explanations of them can be found at [Scikit-Learn's website](https://scikit-learn.org/stable/user_guide.html" \t "_blank).

For this reason, we won't delve too deeply into how they work here, but there will be a brief explanation of how the classifier operates.

#### K-Nearest Neighbors

Credit: [Antti Ajanki AnAj](https://commons.wikimedia.org/w/index.php?curid=2170282)

**K-Nearest Neighbors** operates by checking the distance from some test example to the known values of some training example. The group of data points/class that would give the smallest distance between the training points and the testing point is the class that is selected.

#### Decision Trees

A **Decision Tree Classifier** functions by breaking down a dataset into smaller and smaller subsets based on different criteria. Different sorting criteria will be used to divide the dataset, with the number of examples getting smaller with every division.

Once the network has divided the data down to one example, the example will be put into a class that corresponds to a key. When multiple random forest classifiers are linked together they are called Random Forest Classifiers.

#### Naive Bayes

A **Naive Bayes Classifier** determines the probability that an example belongs to some class, calculating the probability that an event will occur given that some input event has occurred.

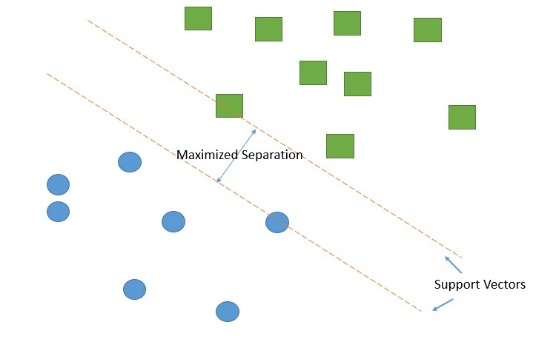
When it does this calculation it is assumed that all the predictors of a class have the same effect on the outcome, that the predictors are independent.

#### Linear Discriminant Analysis

**Linear Discriminant Analysis** works by reducing the dimensionality of the dataset, projecting all of the data points onto a line. Then it combines these points into classes based on their distance from a chosen point or centroid.

Linear discriminant analysis, as you may be able to guess, is a linear classification algorithm and best used when the data has a linear relationship.

#### Support Vector Machines

Credit: [Qluong2016](https://commons.wikimedia.org/wiki/File:Support_vector_machine.jpg)

**Support Vector Machines** work by drawing a line between the different clusters of data points to group them into classes. Points on one side of the line will be one class and points on the other side belong to another class.

The classifier will try to maximize the distance between the line it draws and the points on either side of it, to increase its confidence in which points belong to which class. When the testing points are plotted, the side of the line they fall on is the class they are put in.

#### Logistic Regression

**Logistic Regression** outputs predictions about test data points on a binary scale, zero or one. If the value of something is 0.5 or above, it is classified as belonging to class 1, while below 0.5 if is classified as belonging to 0.

Each of the features also has a label of only 0 or 1. Logistic regression is a linear classifier and therefore used when there is some sort of linear relationship between the data.

<https://analyticsindiamag.com/7-types-classification-algorithms/>

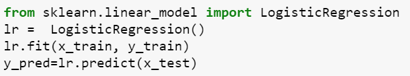
**2 Classification Algorithms (Python)**

**2.1 Logistic Regression**

**Definition:**Logistic regression is a machine learning algorithm for classification. In this algorithm, the probabilities describing the possible outcomes of a single trial are modelled using a logistic function.

**Advantages:** Logistic regression is designed for this purpose (classification), and is most useful for understanding the influence of several independent variables on a single outcome variable.

**Disadvantages:** Works only when the predicted variable is binary, assumes all predictors are independent of each other, and assumes data is free of missing values.

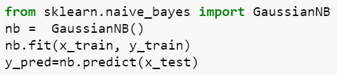
[](https://analyticsindiamag.com/7-types-classification-algorithms/screen-shot-2018-01-19-at-10-52-28-am/)

**2.2 Naïve Bayes**

**Definition:**Naive Bayes algorithm based on Bayes’ theorem with the assumption of independence between every pair of features. Naive Bayes classifiers work well in many real-world situations such as document classification and spam filtering.

**Advantages:**This algorithm requires a small amount of training data to estimate the necessary parameters. Naive Bayes classifiers are extremely fast compared to more sophisticated methods.

**Disadvantages:**Naive Bayes is is known to be a bad estimator.

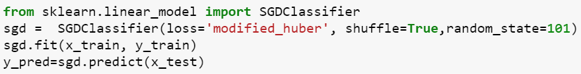
[](https://analyticsindiamag.com/7-types-classification-algorithms/screen-shot-2018-01-19-at-10-52-55-am/)

**2.3 Stochastic Gradient Descent**

**Definition:**Stochastic gradient descent is a simple and very efficient approach to fit linear models. It is particularly useful when the number of samples is very large. It supports different loss functions and penalties for classification.

**Advantages:** Efficiency and ease of implementation.

**Disadvantages:** Requires a number of hyper-parameters and it is sensitive to feature scaling.

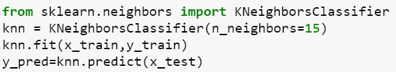
[](https://analyticsindiamag.com/7-types-classification-algorithms/screen-shot-2018-01-19-at-10-53-58-am/)

**2.4 K-Nearest Neighbours**

**Definition:**Neighbours based classification is a type of lazy learning as it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the k nearest neighbours of each point.

**Advantages:**This algorithm is simple to implement, robust to noisy training data, and effective if training data is large.

**Disadvantages:**Need to determine the value of K and the computation cost is high as it needs to computer the distance of each instance to all the training samples.

[](https://analyticsindiamag.com/7-types-classification-algorithms/screen-shot-2018-01-19-at-10-58-19-am/)

**2.5 Decision Tree**

**Definition:** Given a data of attributes together with its classes, a decision tree produces a sequence of rules that can be used to classify the data.

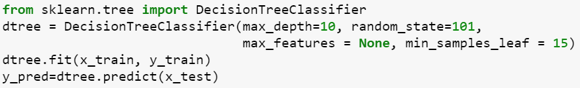
**Advantages:** Decision Tree is simple to understand and visualise, requires little data preparation, and can handle both numerical and categorical data.

**Disadvantages:**Decision tree can create complex trees that do not generalise well, and decision trees can be unstable because small variations in the data might result in a completely different tree being generated.

SEE ALSO

[DEVELOPERS CORNER](https://analyticsindiamag.com/category/learning-corner/)

[**TOP 7 PYTHON LIBRARIES POPULAR IN THE FINTECH SECTOR**](https://analyticsindiamag.com/top-7-python-libraries-popular-in-the-fintech-sector/)

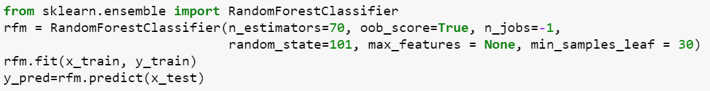
[](https://analyticsindiamag.com/7-types-classification-algorithms/screen-shot-2018-01-19-at-10-59-33-am/)

**2.6 Random Forest**

**Definition:**Random forest classifier is a meta-estimator that fits a number of decision trees on various sub-samples of datasets and uses average to improve the predictive accuracy of the model and controls over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement.

**Advantages:**Reduction in over-fitting and random forest classifier is more accurate than decision trees in most cases.

**Disadvantages:**Slow real time prediction, difficult to implement, and complex algorithm.

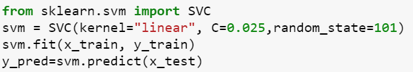
[](https://analyticsindiamag.com/7-types-classification-algorithms/screen-shot-2018-01-19-at-11-00-06-am/)

**2.7 Support Vector Machine**

**Definition:**Support vector machine is a representation of the training data as points in space separated into categories by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

**Advantages:**Effective in high dimensional spaces and uses a subset of training points in the decision function so it is also memory efficient.

**Disadvantages:**The algorithm does not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

[](https://analyticsindiamag.com/7-types-classification-algorithms/screen-shot-2018-01-19-at-11-00-44-am/)

<https://scikit-learn.org/stable/user_guide.html>

1. Describe the difference between bagging and boosting methods, and when to use one or the other.

<https://quantdare.com/what-is-the-difference-between-bagging-and-boosting/>

<https://towardsdatascience.com/decision-tree-ensembles-bagging-and-boosting-266a8ba60fd9>

<https://medium.com/swlh/difference-between-bagging-and-boosting-f996253acd22>

1. Describe 2 regularization techniques for a random forest model
2. <https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/>