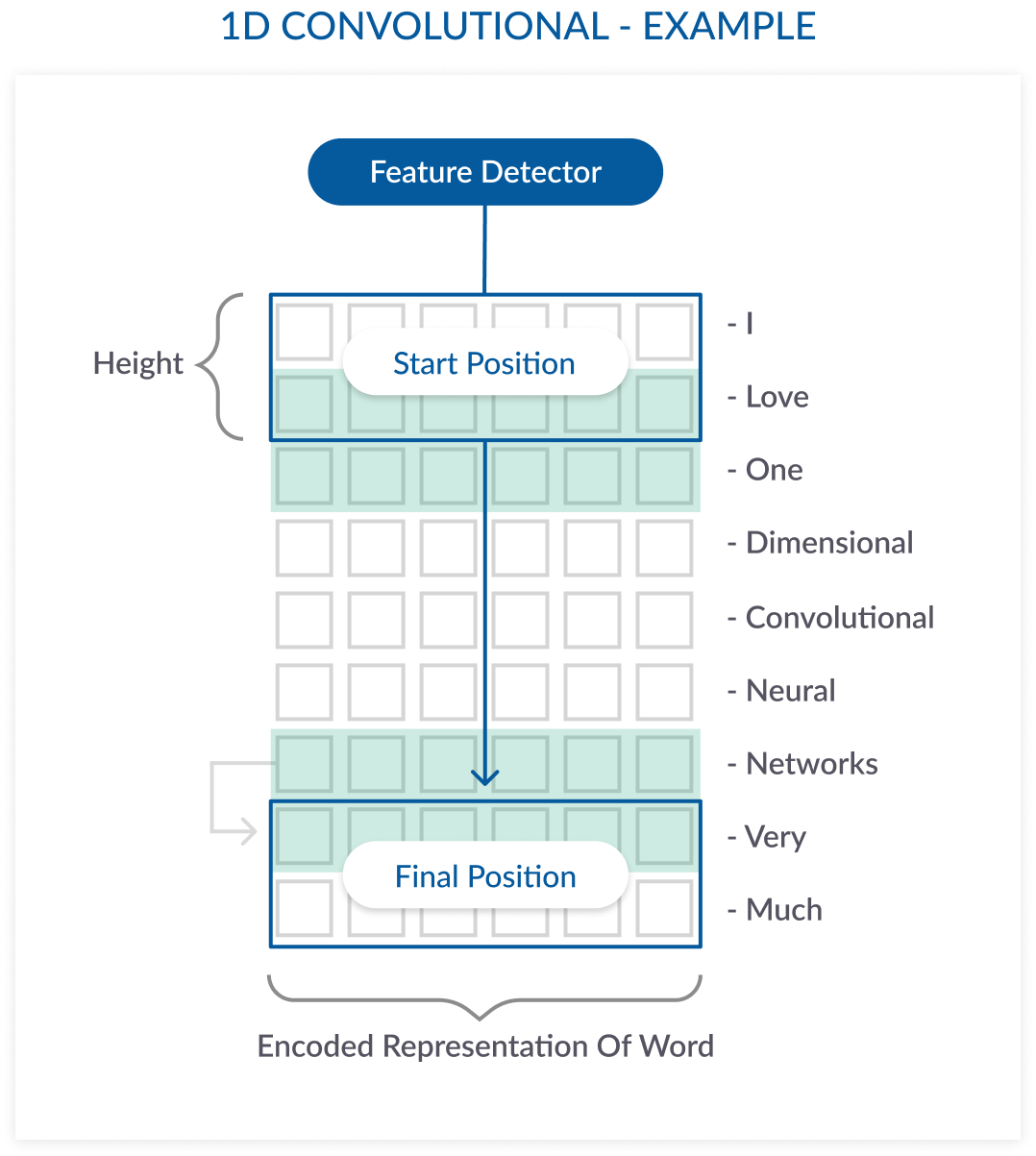
# **1D Convolutional Neural Networks**

**Convolutional Neural Network (CNN)** models were developed for image classification, in which the model accepts a two-dimensional input representing an image’s pixels and color channels, in a process called feature learning.

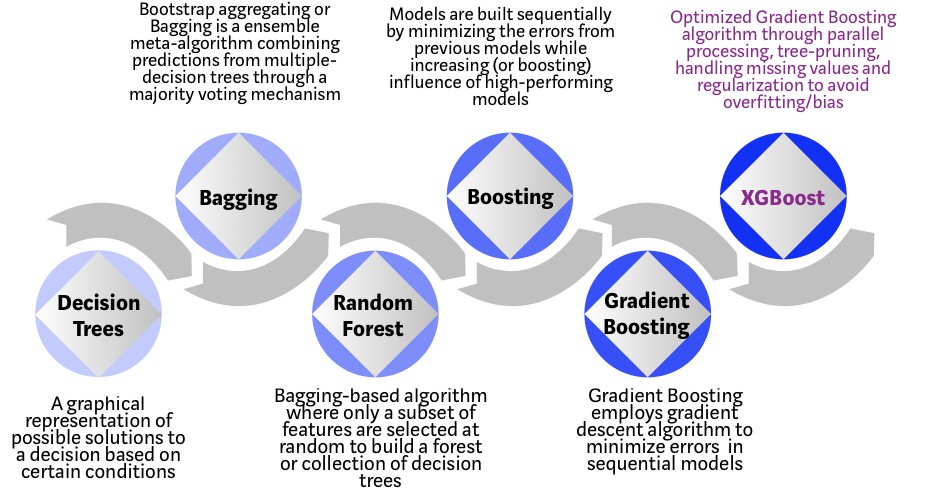
This same process can be applied to one-dimensional sequences of data. The model extracts features from sequences data and maps the internal features of the sequence. A 1D CNN is very effective for deriving features from a fixed-length segment of the overall dataset, where it is not so important where the feature is located in the segment.



In this natural language processing (NLP) example, a sentence is made up of 9 words. Each word is a vector that represents a word. The filter covers at least one word; a height parameter specifies how many words the filter should consider at once. In this example the height is 2, meaning the filter moves 8 times to fully scan the data.

**XGBOOST**

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks. However, when it comes to small-to-medium structured/tabular data, decision tree based algorithms are considered best-in-class right now. Please see the chart below for the evolution of tree-based algorithms over the years.

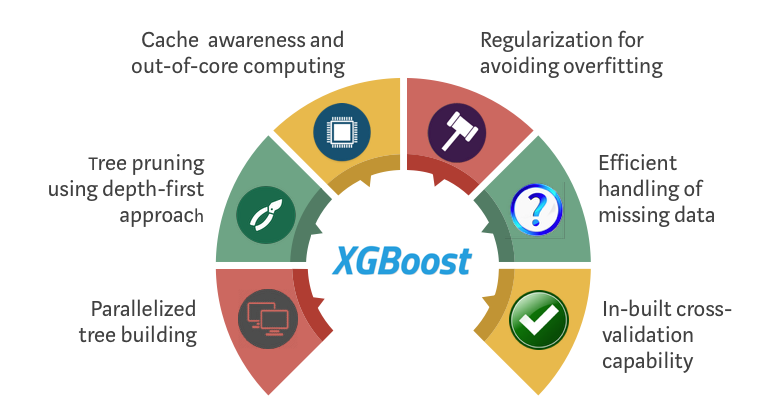


XGBoost algorithm was developed as a research project at the University of Washington. Tianqi Chen and Carlos Guestrin presented their paper at SIGKDD Conference in 2016 and caught the Machine Learning world by fire. Since its introduction, this algorithm has not only been credited with winning numerous Kaggle competitions but also for being the driving force under the hood for several cutting-edge industry applications. As a result, there is a strong community of data scientists contributing to the XGBoost open source projects with ~350 contributors and ~3,600 commits on GitHub. The algorithm differentiates itself in the following ways:

* A wide range of applications: Can be used to solve regression, classification, ranking, and user-defined prediction problems.
* Portability: Runs smoothly on Windows, Linux, and OS X.
* Languages: Supports all major programming languages including C++, Python, R, Java, Scala, and Julia.

Cloud Integration: Supports AWS, Azure, and Yarn clusters and works well with Flink, Spark, and other ecosystems.

**Why does XGBoost perform so well?**

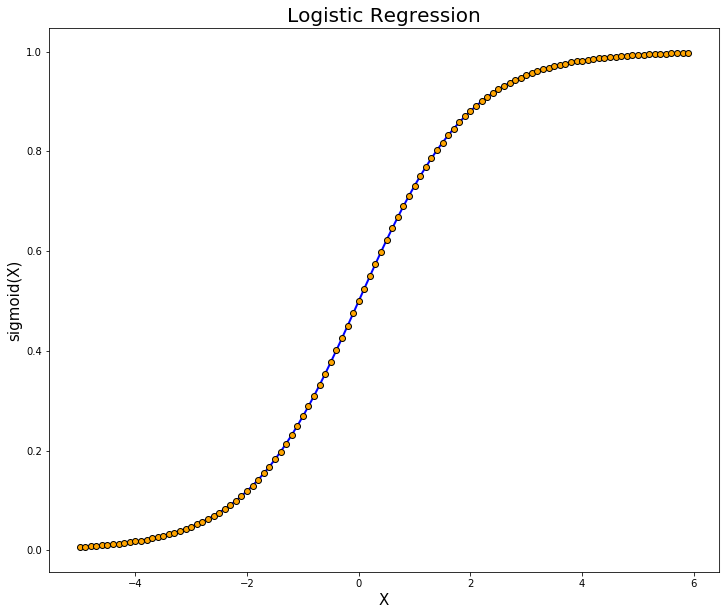
XGBoost and Gradient Boosting Machines (GBMs) are both ensemble tree methods that apply the principle of boosting weak learners (CARTs generally) using the gradient descent architecture. However, XGBoost improves upon the base GBM framework through systems optimization and algorithmic enhancements.

# Logistic Regression

In the Machine Learning, Logistic Regression is a kind of parametric classification model, despite having the word ‘regression’ in its name.

This means that logistic regression models are models that have a certain fixed number of parameters that depend on the number of input features, and they output categorical prediction, like for example if a plant belongs to a certain species or not.

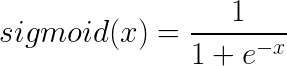
In Logistic Regression, we don’t directly fit a straight line to our data like in linear regression. Instead, we fit a S shaped curve, called Sigmoid, to our observations.



Logistic Regression models are classification models; specifically binary classification models (they can only be used to distinguish between 2 different categories — like if a person is obese or not given its weight, or if a house is big or small given its size). This means that our data has two kinds of observations (Category 1 and Category 2 observations) like we can observe in the figure.

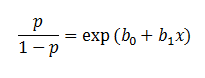
Secondly, as we can see, the Y-axis goes from 0 to 1. This is because the sigmoid function always takes as maximum and minimum these two values, and this fits very well our goal of classifying samples in two different categories. By computing the sigmoid function of X (that is a weighted sum of the input features, just like in Linear Regression), we get a probability (between 0 and 1 obviously) of an observation belonging to one of the two categories.

The formula for the sigmoid function is the following:

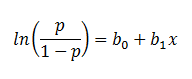




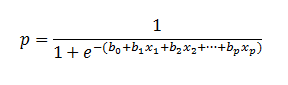
In the logistic regression the constant (b0) moves the curve left and right and the slope (b1) defines the steepness of the curve. By simple transformation, the logistic regression equation can be written in terms of an odds ratio.



Finally, taking the natural log of both sides, we can write the equation in terms of log-odds (logit) which is a linear function of the predictors. The coefficient (b1) is the amount the logit (log-odds) changes with a one unit change in x.



Logistic regression can handle any number of numerical and/or categorical variables.



**Support Vector Machines**

Support vector machines are a core machine learning tech-nology. They have strong theoretical foundations and excel-lent empirical successes. They have been applied to taskssuch as handwritten digit recognition [35], object recogni-tion [25], and text classification [14]

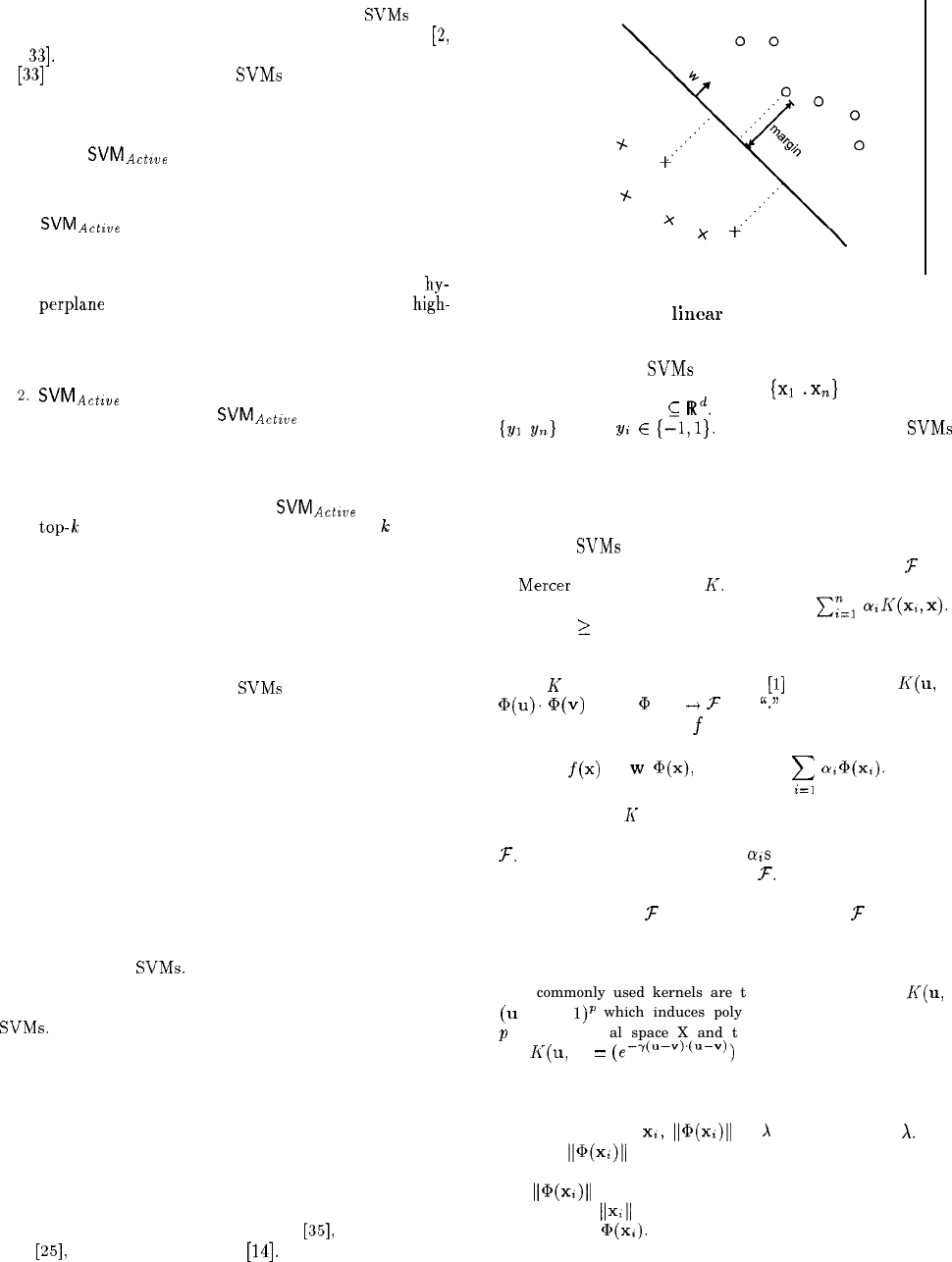


Figure 1: A simple linear Support Vector Machine

We shall consider SVMs in the binary classification setting. We are given training data (x1 . xn} that are vectors in some space .We are also given their labels{y1 yn} where . In their simplest form, SVMs are hyperplanes that separate the training data by a maximal margin (see Fig. 1). All vectors lying on one side of the hyper plane are labeled as -1, and all vectors lying on the other side are labeled as 1. The training instances that lie closest to the hyper plane are called support vectors. More generally, SVMs allow one to project the original training data in space X to a higher dimensional feature space F via **Mercer kernel operator K**.

[35] V. Vapnik. Statistical Learning Theory. Wiley, 1998.

[25 ] C. Papageorgiou, M. Oren, and T. Poggio. A general

framework for object detection. In Proceedings of the

International ConJerence on Computer Vision, 1998.

[14] T. Joachims. Text categorization with support vector machines. In Proceedings of the European C~njerence on Machine Learning. Springer-Verlrtg, 1998.