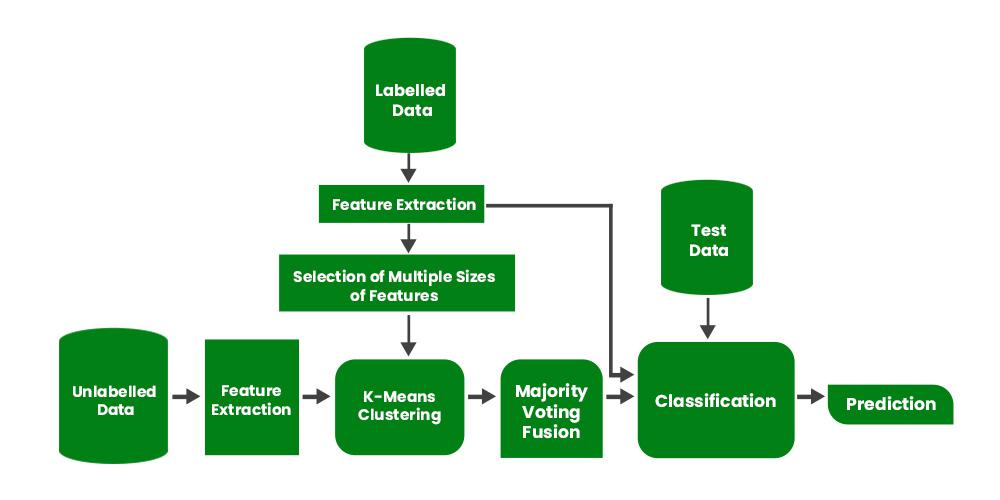
**What is Semi-Supervised Learning?**

Semi-supervised learning is a type of [machine learning](https://www.geeksforgeeks.org/machine-learning/) that falls in between supervised and unsupervised learning. It is a method that uses a small amount of labeled data and a large amount of unlabeled data to train a model. The goal of semi-supervised learning is to learn a function that can accurately predict the output variable based on the input variables, similar to supervised learning. However, unlike supervised learning, the algorithm is trained on a dataset that contains both labeled and unlabeled data.

Semi-supervised learning is particularly useful when there is a large amount of unlabeled data available, but it’s too expensive or difficult to label all of it.



Semi-Supervised Learning Flow Chart

Intuitively, one may imagine the three types of learning algorithms as Supervised learning where a student is under the supervision of a teacher at both home and school, Unsupervised learning where a student has to figure out a concept himself and Semi-Supervised learning where a teacher teaches a few concepts in class and gives questions as homework which are based on similar concepts.

**Examples of Semi-Supervised Learning**

* [**Text classification**](https://www.geeksforgeeks.org/sentiment-classification-using-bert/): In text classification, the goal is to classify a given text into one or more predefined categories. Semi-supervised learning can be used to train a text classification model using a small amount of labeled data and a large amount of unlabeled text data.
* [**Image classification**](https://www.geeksforgeeks.org/python-image-classification-using-keras/): In image classification, the goal is to classify a given image into one or more predefined categories. Semi-supervised learning can be used to train an image classification model using a small amount of labeled data and a large amount of unlabeled image data.
* [**Anomaly** **detection**](https://www.geeksforgeeks.org/machine-learning-for-anomaly-detection/): In anomaly detection, the goal is to detect patterns or observations that are unusual or different from the norm

**Assumptions followed by Semi-Supervised Learning**

A Semi-Supervised algorithm assumes the following about the data

1. **Continuity Assumption:** The algorithm assumes that the points which are closer to each other are more likely to have the same output label.
2. **Cluster Assumption:** The data can be divided into discrete clusters and points in the same cluster are more likely to share an output label.
3. **Manifold Assumption:** The data lie approximately on a manifold of a much lower dimension than the input space. This assumption allows the use of distances and densities which are defined on a manifold.

**Applications of Semi-Supervised Learning**

1. **Speech Analysis:** Since labeling audio files is a very intensive task, Semi-Supervised learning is a very natural approach to solve this problem.
2. **Internet Content Classification:** Labeling each webpage is an impractical and unfeasible process and thus uses Semi-Supervised learning algorithms. Even the Google search algorithm uses a variant of Semi-Supervised learning to rank the relevance of a webpage for a given query.
3. **Protein Sequence Classification:** Since DNA strands are typically very large in size, the rise of Semi-Supervised learning has been imminent in this field.

**Disadvantages of Semi-Supervised Learning**

The most basic disadvantage of any [**Supervised Learning**](https://www.geeksforgeeks.org/ml-types-learning-supervised-learning/) algorithm is that the dataset has to be hand-labeled either by a Machine Learning Engineer or a Data Scientist. This is a very *costly process*, especially when dealing with large volumes of data. The most basic disadvantage of any [**Unsupervised Learning**](https://www.geeksforgeeks.org/supervised-unsupervised-learning/) is that its **application spectrum is limited**.

To counter these disadvantages, the concept of **Semi-Supervised Learning** was introduced. In this type of learning, the algorithm is trained upon a combination of labeled and unlabelled data. Typically, this combination will contain a very small amount of labeled data and a very large amount of unlabelled data. The basic procedure involved is that first, the programmer will cluster similar data using an unsupervised learning algorithm and then use the existing labeled data to label the rest of the unlabelled data. The typical use cases of such type of algorithm have a common property among them – The acquisition of unlabelled data is relatively cheap while labeling the said data is very expensive.

Genetic Algorithms

Genetic Algorithms(GAs) are adaptive heuristic search algorithms that belong to the larger part of evolutionary algorithms. Genetic algorithms are based on the ideas of natural selection and genetics. These are intelligent exploitation of random search provided with historical data to direct the search into the region of better performance in solution space. **They are commonly used to generate high-quality solutions for optimization problems and search problems.**

**Genetic algorithms simulate the process of natural selection** which means those species who can adapt to changes in their environment are able to survive and reproduce and go to next generation. In simple words, they simulate “survival of the fittest” among individual of consecutive generation for solving a problem. **Each generation consist of a population of individuals** and each individual represents a point in search space and possible solution. Each individual is represented as a string of character/integer/float/bits. This string is analogous to the Chromosome.

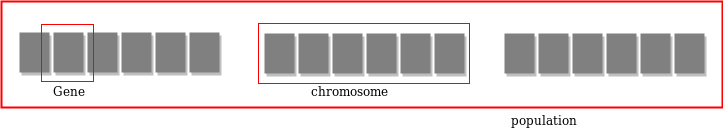
**Foundation of Genetic Algorithms**

Genetic algorithms are based on an analogy with genetic structure and behaviour of chromosomes of the population. Following is the foundation of GAs based on this analogy –

1. Individual in population compete for resources and mate
2. Those individuals who are successful (fittest) then mate to create more offspring than others
3. Genes from “fittest” parent propagate throughout the generation, that is sometimes parents create offspring which is better than either parent.
4. Thus each successive generation is more suited for their environment.

**Search space**

The population of individuals are maintained within search space. Each individual represents a solution in search space for given problem. Each individual is coded as a finite length vector (analogous to chromosome) of components. These variable components are analogous to Genes. Thus a chromosome (individual) is composed of several genes (variable components).



**Fitness Score**

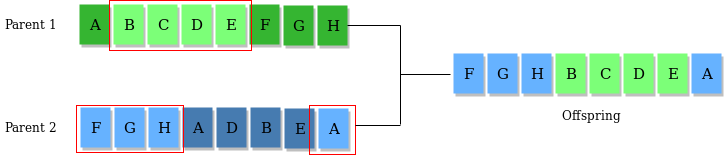
A Fitness Score is given to each individual which **shows the ability of an individual to “compete”**. The individual having optimal fitness score (or near optimal) are sought.

The GAs maintains the population of n individuals (chromosome/solutions) along with their fitness scores.The individuals having better fitness scores are given more chance to reproduce than others. The individuals with better fitness scores are selected who mate and produce **better offspring** by combining chromosomes of parents. The population size is static so the room has to be created for new arrivals. So, some individuals die and get replaced by new arrivals eventually creating new generation when all the mating opportunity of the old population is exhausted. It is hoped that over successive generations better solutions will arrive while least fit die.

Each new generation has on average more “better genes” than the individual (solution) of previous generations. Thus each new generations have better **“partial solutions”** than previous generations. Once the offspring produced having no significant difference from offspring produced by previous populations, the population is converged. The algorithm is said to be converged to a set of solutions for the problem.

**Operators of Genetic Algorithms**

Once the initial generation is created, the algorithm evolves the generation using following operators –   
**1) Selection Operator:** The idea is to give preference to the individuals with good fitness scores and allow them to pass their genes to successive generations.   
**2) Crossover Operator:** This represents mating between individuals. Two individuals are selected using selection operator and crossover sites are chosen randomly. Then the genes at these crossover sites are exchanged thus creating a completely new individual (offspring). For example –



**3) Mutation Operator:** The key idea is to insert random genes in offspring to maintain the diversity in the population to avoid premature convergence. For example – 



The whole algorithm can be summarized as –

1) Randomly initialize populations p

2) Determine fitness of population

3) Until convergence repeat:

a) Select parents from population

b) Crossover and generate new population

c) Perform mutation on new population

d) Calculate fitness for new population

**Example problem and solution using Genetic Algorithms**

Given a target string, the goal is to produce target string starting from a random string of the same length. In the following implementation, following analogies are made –

* Characters A-Z, a-z, 0-9, and other special symbols are considered as genes
* A string generated by these characters is considered as chromosome/solution/Individual

**Fitness score** is the number of characters which differ from characters in target string at a particular index. So individual having lower fitness value is given more preference.

**Ml models**

o begin with it is important to understand the objective behind these tools, which is primarily learning from data. Both these approaches aim to use the data generated to understand the underlying phenomena. It is like two games being played on the same board but with different rules.

[Introducing HeavyRF    |    Watch: Telco Digital Twins Webinar   |    Learn More](https://www.heavy.ai/product/heavyrf" \t "_blank)

[[](https://www.heavy.ai/)](https://www.heavy.ai/)

[GET HEAVY.AI FREE](https://www.heavy.ai/product/downloads)

# Statistical Modeling

## Statistical Modeling Definition

Statistical modeling is the use of mathematical models and statistical assumptions to generate sample data and make predictions about the real world. A statistical model is a collection of probability distributions on a set of all possible outcomes of an experiment.

‍

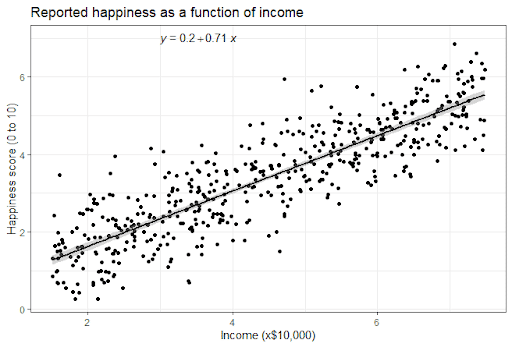


Image from [Scribbr](https://www.scribbr.com/statistics/simple-linear-regression/)

‍

##### FAQs

‍

## What is Statistical Modeling?

Statistical modeling refers to the [data science](https://www.heavy.ai/learn/data-science) process of applying statistical analysis to datasets. A statistical model is a mathematical relationship between one or more random variables and other non-random variables. The application of statistical modeling to raw data helps data scientists approach data analysis in a strategic manner, providing intuitive visualizations that aid in identifying relationships between variables and [making predictions](https://www.heavy.ai/technical-glossary/predictive-analytics).  
‍

Common data sets for statistical analysis include Internet of Things (IoT) sensors, census data, public health data, social media data, imagery data, and other [public sector](https://www.heavy.ai/resources/solution-brief/public-sector-analytics/lp) data that benefit from real-world predictions.

‍

## Statistical Modeling Techniques

The first step in developing a statistical model is gathering data, which may be sourced from spreadsheets, databases, data lakes, or the cloud. The most common statistical modeling methods for analyzing this data are categorized as either supervised learning or unsupervised learning. Some popular statistical model examples include logistic regression, time-series, clustering, and decision trees.   
‍

Supervised learning techniques include regression models and classification models:

* **Regression model**: a type of predictive statistical model that analyzes the relationship between a dependent and an independent variable. Common regression models include logistic, polynomial, and linear regression models. Use cases include forecasting, time series modeling, and discovering the causal effect relationship between variables.
* **Classification model**: a type of machine learning in which an algorithm analyzes an existing, large and complex set of known data points as a means of understanding and then appropriately classifying the data; common models include models include decision trees, Naive Bayes, nearest neighbor, random forests, and neural networking models, which are typically used in Artificial Intelligence.  
  ‍

Unsupervised learning techniques include clustering algorithms and association rules:

* **K-means clustering**: aggregates a specified number of data points into a specific number of groupings based on certain similarities.
* **Reinforcement learning**: an area of deep learning that concerns models iterating over many attempts, rewarding moves that produce favorable outcomes and penalizing steps that produce undesired outcomes, therefore training the algorithm to learn the optimal process.  
  ‍

There are three main types of statistical models: parametric, nonparametric, and semiparametric:

* **Parametric**: a family of probability distributions that has a finite number of parameters.
* **Nonparametric**: models in which the number and nature of the parameters are flexible and not fixed in advance.**‍**
* **Semiparametric**: the parameter has both a finite-dimensional component (parametric) and an infinite-dimensional component (nonparametric).

‍

## How to Build Statistical Models

The first step in building a statistical model is knowing how to choose a statistical model. Choosing the best statistical model is dependent upon several different variables. Is the purpose of the analysis to answer a very specific question, or solely to make predictions from a set of variables? How many explanatory and dependent variables are there? What is the shape of the relationships between dependent and explanatory variables? How many parameters will be included in the model? Once these questions are answered, the appropriate model can be selected.   
‍

Once a statistical model is selected, it must be built. Best practices for how to make a statistical model include:  
‍

* Start with univariate descriptives and graphs. Visualizing the data helps with identifying errors, understanding the variables you’re working with, how they look, how they are behaving and why.
* Build predictors in theoretically distinct sets first in order to observe how related variables work together, and then the outcome once the sets are combined.
* Next, run bivariate descriptives with graphs in order to visualize and understand how each potential predictor relates individually to every other predictor and to the outcome.
* Frequently record, compare and interpret results from models run with and without control variables.
* Eliminate non-significant interactions first; any variable involved in a significant interaction must be included in the model by itself.
* While identifying the many existing relationships between variables, and categorizing and testing every possible predictor, be sure not to lose sight of the research question.

‍

## Statistical Modeling vs Mathematical Modeling

Much like statistical modeling, mathematical modeling translates real-world problems into tractable mathematical formulations whose analysis provides insight, results and direction useful for the originating application. However, unlike statistical modeling, mathematical modeling involves static models that represent a real-world phenomenon in mathematical form. Once a mathematical model is formulated, it does not necessitate change. Statistical models are flexible and, with the aid of machine learning, can incorporate new, emerging patterns and trends, and will adjust with the introduction of new data.

‍

## Machine Learning vs Statistical Modeling

Machine learning is a subfield of computer science and artificial intelligence that involves building systems that can learn from data rather than explicitly programmed instructions. Machine learning models seek out patterns hidden in data independent of all assumptions, therefore predictive power is typically very strong. Machine learning requires little human input and does well with large numbers of attributes and observations.  
‍

Statistical modeling is a subfield of mathematics that seeks out relationships between variables in order to predict an outcome. Statistical models are based on coefficient estimation, are typically applied to smaller sets of data with fewer attributes, and require the human designer to understand the relationships between variables before inputting.

‍

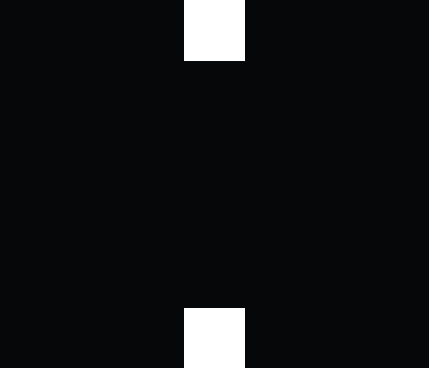
## Statistical Modeling Software

Statistical modeling software are specialized computer programs that help gather, organize, analyze, interpret and statistically design data. Advanced statistics software should provide data mining, data importation, analysis and reporting, automated data modeling and deployment, data visualization, multi-platform support, prediction capabilities, and an intuitive user interface with statistical features ranging from basic tabulations to multilevel models. Statistical software is available as proprietary, open-source, public domain, and freeware.

‍

## Does HEAVY.AI a Statistical Modeling Solution?

Statistical modeling serves as one of the solutions for the data discovery challenge facing big data management systems. HEAVY.AI's [Data Science Platform](https://www.heavy.ai/platform/data-science-platform) provides an always-on dashboard for monitoring the health of statistical models in which the user can visualize predictions alongside actual outcomes and see how predications diverge from real life.



Product