Contents

[1. Machine Learning 1](#_Toc508546519)

[2. Supervised learning: 1](#_Toc508546520)

[3. Unsupervised learning: 1](#_Toc508546521)

[4. Probability 1](#_Toc508546522)

[4.1. Probability Theory 1](#_Toc508546523)

[4.2. Probability axioms 2](#_Toc508546524)

[4.3. Applications 2](#_Toc508546525)

[5. Bayes' theorem 3](#_Toc508546526)

[6. Classification 4](#_Toc508546527)

[6.1. Naive Bayes classifier 4](#_Toc508546528)

[6.1.1. Intoroduction 5](#_Toc508546529)

[6.2. Gaussian naive Bayes 5](#_Toc508546530)

[6.3. Decision Tree 6](#_Toc508546531)

# Machine Learning

“Machine learning is a field of computer science that gives computer system the ability to learn with data, without being explicitly programmed.”

In very simple terms machine learning is learning by examples

(Here learning means progressively improve performance on a specific task.)

Machine learning explores the study and construction of algorithms that can learn from data and can make predictions on data. Example applications include email filtering, face detection, optical character recognition (OCR), and computer vision. Machine learning is closely related to (and often overlaps with) [computational statistics](https://en.wikipedia.org/wiki/Computational_statistics), which also focuses on prediction-making through the use of computers

Some important terms in ML are:

## Labels

A **label** is the thing we're predicting—the y variable in simple linear regression. The label could be the future price of wheat, the kind of animal shown in a picture, the meaning of an audio clip, or just about anything.

## Features

A feature is an input variable—the x variable in simple linear regression. A simple machine learning project might use a single feature, while a more sophisticated machine learning project could use millions of features, specified as:

{ x1, x2, x3, ……… xn }

n the spam detector example, the features could include the following:

* words in the email text
* sender's address
* time of day the email was sent
* email contains the phrase "one weird trick."

## Examples

An **example** is a particular instance of data, **x**. (We put **x** in boldface to indicate that it is a vector.) We break examples into two categories:

labeled examples

unlabeled examples

A **labeled example** includes both feature(s) and the label. That is:

  labeled examples: {features, label}: (x, y)

Use labeled examples to **train** the model. In our spam detector example, the labeled examples would be individual emails that users have explicitly marked as "spam" or "not spam."

For example, the following table shows 5 labeled examples from a [data set](https://developers.google.com/machine-learning/crash-course/california-housing-data-description) containing information about housing prices in California:

|  |  |  |  |
| --- | --- | --- | --- |
| housingMedianAge (feature) | totalRooms (feature) | totalBedrooms (feature) | medianHouseValue (label) |
| 15 | 5612 | 1283 | 66900 |
| 19 | 7650 | 1901 | 80100 |
| 17 | 720 | 174 | 85700 |
| 14 | 1501 | 337 | 73400 |
| 20 | 1454 | 326 | 65500 |

An **unlabeled example** contains features but not the label. That is:

  unlabeled examples: {features, ?}: (x, ?)

Once we've trained our model with labeled examples, we use that model to predict the label on unlabeled examples. In the spam detector, unlabeled examples are new emails that humans haven't yet labeled.

## Models

A model defines the relationship between features and label. For example, a spam detection model might associate certain features strongly with "spam". Let's highlight two phases of a model's life:

* **Training** means creating or **learning** the model. That is, you show the model labeled examples and enable the model to gradually learn the relationships between features and label.
* **Inference** means applying the trained model to unlabeled examples. That is, you use the trained model to make useful predictions (y'). For example, during inference, you can predict medianHouseValue for new unlabeled examples.

## Regression vs. classification

A **regression** model predicts continuous values. For example, regression models make predictions that answer questions like the following:

* What is the value of a house in California?
* What is the probability that a user will click on this ad?

A **classification** model predicts discrete values. For example, classification models make predictions that answer questions like the following:

* Is a given email message spam or not spam?
* Is this an image of a dog, a cat, or a hamster?

Machine learning tasks are typically classified into two broad categories, depending on whether there is a learning "signal" or "feedback" available to a learning system:

# Supervised learning:

The computer is presented with example inputs and their desired outputs and the goal is to learn a general rule that maps inputs to outputs. Once the model is trained on the existing input and output data, same can be used to predict the output of input data seen never before.

For example if sample data for the Housing Price in a city is given with various features and known price of the house, then if for a new house its price can be predicted supervised learning model based on the features of the house.

Supervised learning is a type of machine learning where we feed our system what we call labeled examples. Labeled examples means we give the system some information that it uses to understand the data that we're giving it. So for example if we were building our system to recognize my face, we would

give it some pictures of my face and some pictures of other people's faces. And we would actually tell the system these pictures are of David and these pictures are of somebody else. And it would use that information to develop its understanding over time .Or in education; we would give it examples of students and tell it whether or not each of those students actually ended up graduating. So the system is using some information that we give it about the success or failure of those different examples to construct its understanding about what leads to that success or failure.

# [Unsupervised learning](https://en.wikipedia.org/wiki/Unsupervised_learning):

No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

# Probability

Probability is the measure of the likelihood that an event will occur. See glossary of probability and statistics. Probability is quantified as a number between 0 and 1, where, loosely speaking, 0 indicates impossibility and 1 indicates certainty. The higher the probability of an event, the more likely it is that the event will occur. A simple example is the tossing of a fair (unbiased) coin. Since the coin is fair, the two outcomes ("heads" and "tails") are both equally probable; the probability of "heads" equals the probability of "tails"; and since no other outcomes are possible, the probability of either "heads" or "tails" is 1/2 (which could also be written as 0.5 or 50%).

## Probability Theory

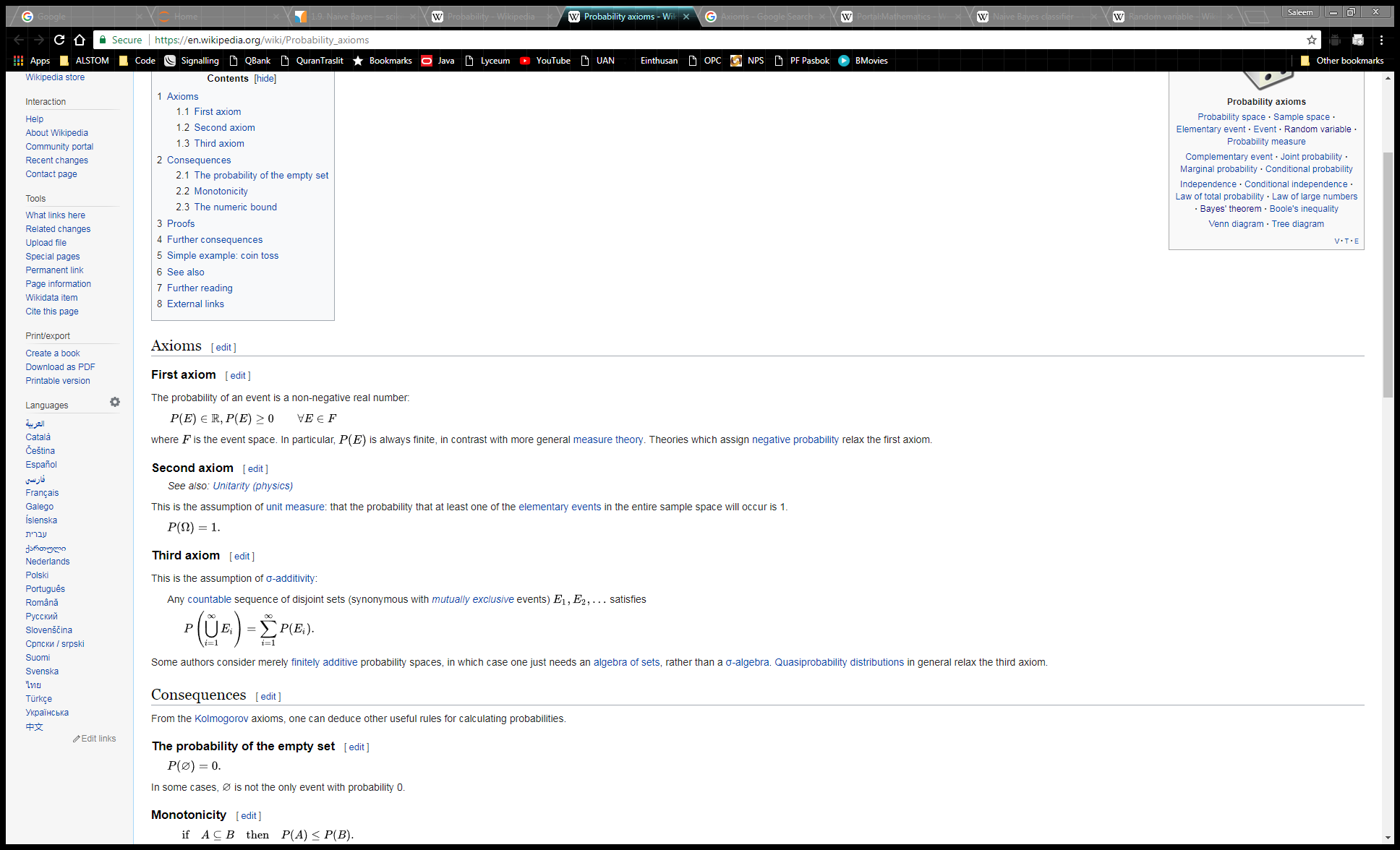
**Probability theory** is the branch of [mathematics](https://en.wikipedia.org/wiki/Mathematics) concerned with [probability](https://en.wikipedia.org/wiki/Probability). Although there are several different [probability interpretations](https://en.wikipedia.org/wiki/Probability_interpretations), probability theory treats the concept in a rigorous mathematical manner by expressing it through a set of [axioms](https://en.wikipedia.org/wiki/Axioms_of_probability). Typically these axioms formalize probability in terms of a [probability space](https://en.wikipedia.org/wiki/Probability_space), which assigns a [measure](https://en.wikipedia.org/wiki/Measure_(mathematics)) taking values between 0 and 1, termed the [probability measure](https://en.wikipedia.org/wiki/Probability_measure), to a set of outcomes called the [sample space](https://en.wikipedia.org/wiki/Sample_space). Any specified subset of these outcomes is called an [event](https://en.wikipedia.org/wiki/Event_(probability_theory)).

Central subjects in probability theory include discrete and continuous [random variables](https://en.wikipedia.org/wiki/Random_variable), [probability distributions](https://en.wikipedia.org/wiki/Probability_distributions), and [stochastic processes](https://en.wikipedia.org/wiki/Stochastic_process), which provide mathematical abstractions of [non-deterministic](https://en.wikipedia.org/wiki/Determinism) or uncertain processes or measured [quantities](https://en.wikipedia.org/wiki/Quantity) that may either be single occurrences or evolve over time in a random fashion.

Although it is not possible to perfectly predict random events, much can be said about their behavior. Two major results in probability theory describing such behavior are the [law of large numbers](https://en.wikipedia.org/wiki/Law_of_large_numbers) and the [central limit theorem](https://en.wikipedia.org/wiki/Central_limit_theorem).

As a mathematical foundation for [statistics](https://en.wikipedia.org/wiki/Statistics), probability theory is essential to many human activities that involve quantitative analysis of data.[[1]](https://en.wikipedia.org/wiki/Probability_theory#cite_note-1) Methods of probability theory also apply to descriptions of complex systems given only partial knowledge of their state, as in [statistical mechanics](https://en.wikipedia.org/wiki/Statistical_mechanics). A great discovery of twentieth-century [physics](https://en.wikipedia.org/wiki/Physics) was the probabilistic nature of physical phenomena at atomic scales, described in [quantum mechanics](https://en.wikipedia.org/wiki/Quantum_mechanics).

## Probability axioms



## Applications

Probability theory is applied in everyday life in risk assessment and modeling. The insurance industry and markets use actuarial science to determine pricing and make trading decisions. Governments apply probabilistic methods in environmental regulation, entitlement analysis (Reliability theory of aging and longevity), and financial regulation.

A good example of the use of probability theory in equity trading is the effect of the perceived probability of any widespread Middle East conflict on oil prices, which have ripple effects in the economy as a whole. An assessment by a commodity trader that a war is more likely can send that commodity's prices up or down, and signals other traders of that opinion. Accordingly, the probabilities are neither assessed independently nor necessarily very rationally. The theory of behavioral finance emerged to describe the effect of such groupthink on pricing, on policy, and on peace and conflict.

In addition to financial assessment, probability can be used to analyze trends in biology (e.g. disease spread) as well as ecology (e.g. biological Punnett squares). As with finance, risk assessment can be used as a statistical tool to calculate the likelihood of undesirable events occurring and can assist with implementing protocols to avoid encountering such circumstances. Probability is used to design games of chance so that casinos can make a guaranteed profit, yet provide payouts to players that are frequent enough to encourage continued play.

The discovery of rigorous methods to assess and combine probability assessments has changed society. It is important for most citizens to understand how probability assessments are made, and how they contribute to decisions.

Another significant application of probability theory in everyday life is reliability. Many consumer products, such as automobiles and consumer electronics, use reliability theory in product design to reduce the probability of failure. Failure probability may influence a manufacturer's decisions on a product's warranty.

The cache language model and other statistical language models that are used in natural language processing are also examples of applications of probability theory.

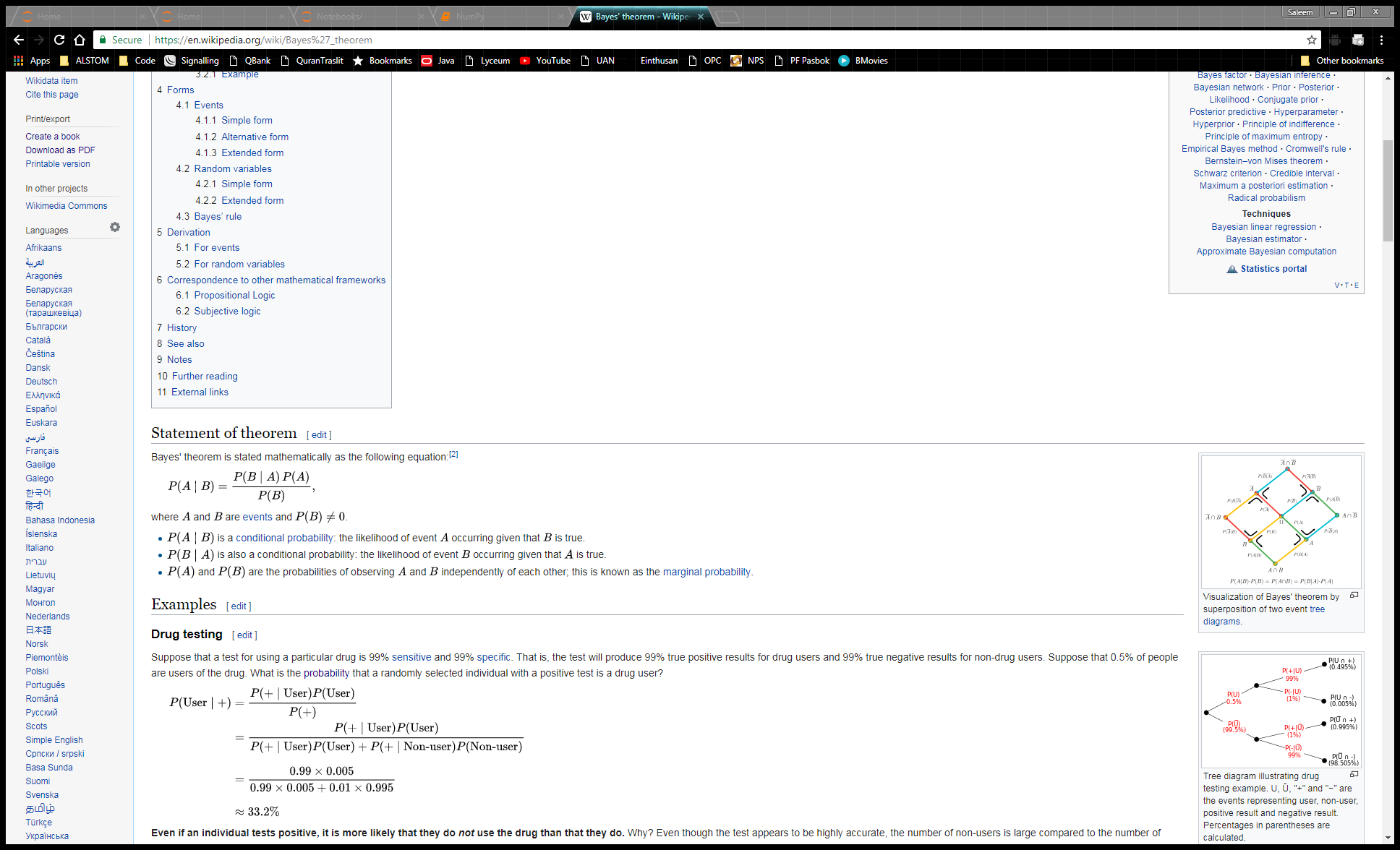
# Bayes' theorem

<https://en.wikipedia.org/wiki/Bayes%27_theorem>

In [probability theory](https://en.wikipedia.org/wiki/Probability_theory) and [statistics](https://en.wikipedia.org/wiki/Statistics), Bayes’ theorem (alternatively Bayes’ law or Bayes' rule) describes the [probability](https://en.wikipedia.org/wiki/Probability) of an [event](https://en.wikipedia.org/wiki/Event_(probability_theory)), based on prior knowledge of conditions that might be related to the event. For example, if cancer is related to age, then, using Bayes’ theorem, a person’s age can be used to more accurately assess the probability that they have cancer, compared to the assessment of the probability of cancer made without knowledge of the person's age.

One of the many applications of Bayes' theorem is [Bayesian inference](https://en.wikipedia.org/wiki/Bayesian_inference), a particular approach to [statistical inference](https://en.wikipedia.org/wiki/Statistical_inference). When applied, the probabilities involved in Bayes' theorem may have different [probability interpretations](https://en.wikipedia.org/wiki/Probability_interpretation). With the [Bayesian probability](https://en.wikipedia.org/wiki/Bayesian_probability) interpretation the theorem expresses how a subjective degree of belief should rationally change to account for availability of related evidence. Bayesian inference is fundamental to [Bayesian statistics](https://en.wikipedia.org/wiki/Bayesian_statistics).

Bayes’ theorem is named after Reverend [Thomas Bayes](https://en.wikipedia.org/wiki/Thomas_Bayes) ([/beɪz/](https://en.wikipedia.org/wiki/Help:IPA/English); 1701–1761), who first provided an equation that allows new evidence to update beliefs in his [An Essay towards solving a Problem in the Doctrine of Chances](https://en.wikipedia.org/wiki/An_Essay_towards_solving_a_Problem_in_the_Doctrine_of_Chances) (1763). It was further developed by [Pierre-Simon Laplace](https://en.wikipedia.org/wiki/Pierre-Simon_Laplace), who first published the modern formulation in his 1812 "[Théorieanalytique des probabilités](https://en.wikipedia.org/wiki/Pierre-Simon_Laplace" \l "Analytic_theory_of_probabilities" \o "Pierre-Simon Laplace)". [Sir Harold Jeffreys](https://en.wikipedia.org/wiki/Harold_Jeffreys) put Bayes’ algorithm and Laplace's formulation on an axiomatic basis. Jeffreys wrote that Bayes' theorem "is to the theory of probability what the [Pythagorean theorem](https://en.wikipedia.org/wiki/Pythagorean_theorem) is to geometry".





# Classification

## Naive Bayes classifier

In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independenceassumptions between the features.

Naive Bayes has been studied extensively since the 1950s. It was introduced under a different name into the text retrieval community in the early 1960s,[1]:488 and remains a popular (baseline) method for text categorization, the problem of judging documents as belonging to one category or the other (such as spam or legitimate, sports or politics, etc.) with word frequencies as the features. With appropriate pre-processing, it is competitive in this domain with more advanced methods including support vector machines. It also finds application in automatic medical diagnosis.

Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression,[1]:718 which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

In the statistics and computer science literature, naive Bayes models are known under a variety of names, including simple Bayes and independence Bayes. All these names reference the use of Bayes' theorem in the classifier's decision rule, but naive Bayes is not (necessarily) a Bayesian method.

### Intoroduction

<https://en.wikipedia.org/wiki/Naive_Bayes_classifier>

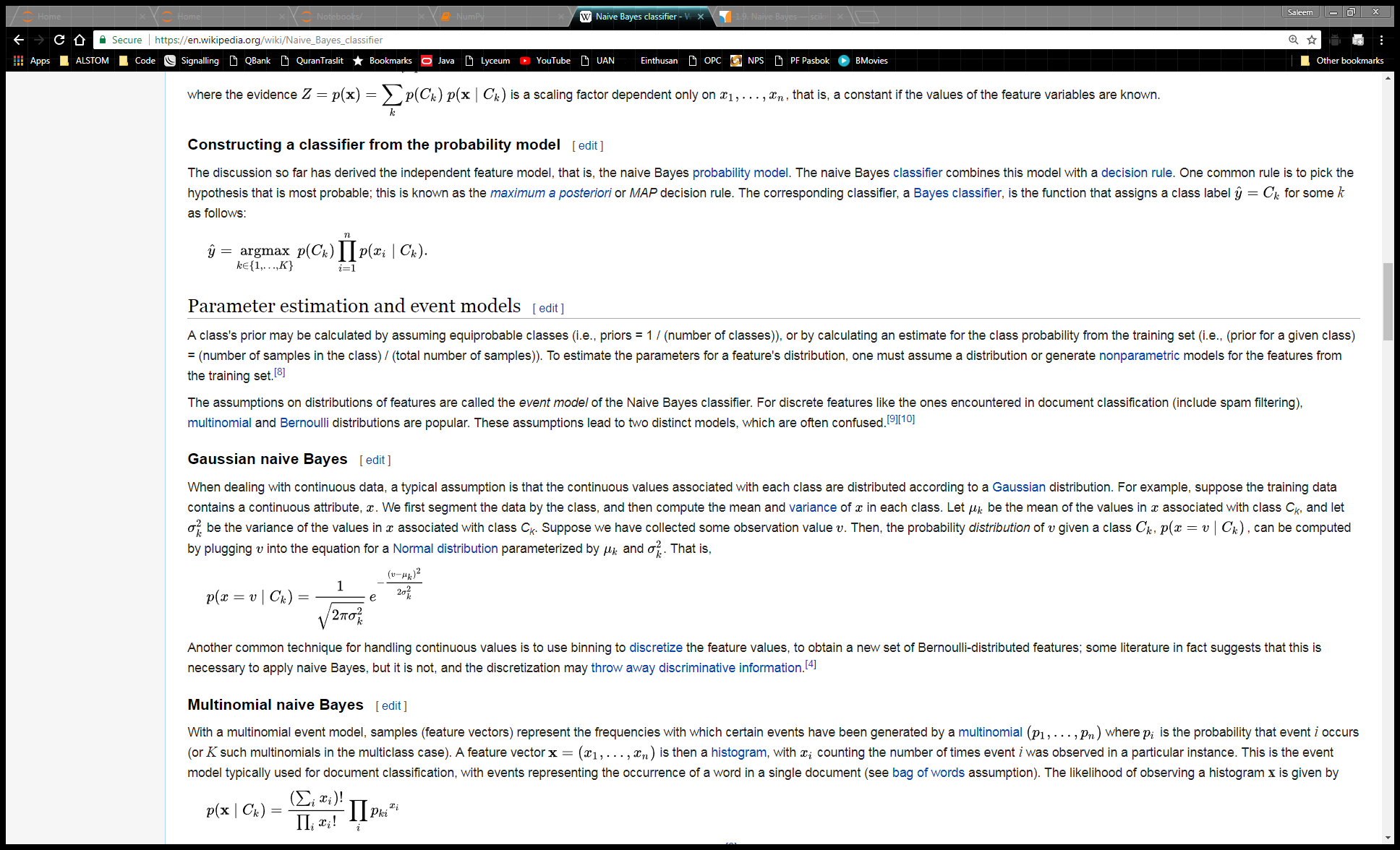
Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features.

For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods.

Despite their naive design and apparently oversimplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations. In 2004, an analysis of the Bayesian classification problem showed that there are sound theoretical reasons for the apparently implausible efficacy of naive Bayes classifiers. Still, a comprehensive comparison with other classification algorithms in 2006 showed that Bayes classification is outperformed by other approaches, such as boosted trees or random forests.

An advantage of naive Bayes is that it only requires a small number of training data to estimate the parameters necessary for classification.

## Gaussian naive Bayes



## Decision Tree

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

#### Advantages

* Simple to understand and to interpret. Trees can be visualised.
* Requires little data preparation. Other techniques often require data normalisation, dummy variables need to be created and blank values to be removed. Note however that this module does not support missing values.
* The cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree.
* Able to handle both numerical and categorical data. Other techniques are usually specialised in analysing datasets that have only one type of variable. See [algorithms](http://scikit-learn.org/stable/modules/tree.html#tree-algorithms) for more information.
* Able to handle multi-output problems.
* Uses a white box model. If a given situation is observable in a model, the explanation for the condition is easily explained by boolean logic. By contrast, in a black box model (e.g., in an artificial neural network), results may be more difficult to interpret.
* Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model.
* Performs well even if its assumptions are somewhat violated by the true model from which the data were generated.

#### Disadvantages

* .Decision-tree learners can create over-complex trees that do not generalise the data well. This is called overfitting. Mechanisms such as pruning (not currently supported), setting the minimum number of samples required at a leaf node or setting the maximum depth of the tree are necessary to avoid this problem.
* Decision trees can be unstable because small variations in the data might result in a completely different tree being generated. This problem is mitigated by using decision trees within an ensemble.
* The problem of learning an optimal decision tree is known to be NP-complete under several aspects of optimality and even for simple concepts. Consequently, practical decision-tree learning algorithms are based on heuristic algorithms such as the greedy algorithm where locally optimal decisions are made at each node. Such algorithms cannot guarantee to return the globally optimal decision tree. This can be mitigated by training multiple trees in an ensemble learner, where the features and samples are randomly sampled with replacement.
* There are concepts that are hard to learn because decision trees do not express them easily, such as XOR, parity or multiplexer problems.
* Decision tree learners create biased trees if some classes dominate. It is therefore recommended to balance the dataset prior to fitting with the decision tree.