Evaluating Hypothesis:

What is Hypothesis?

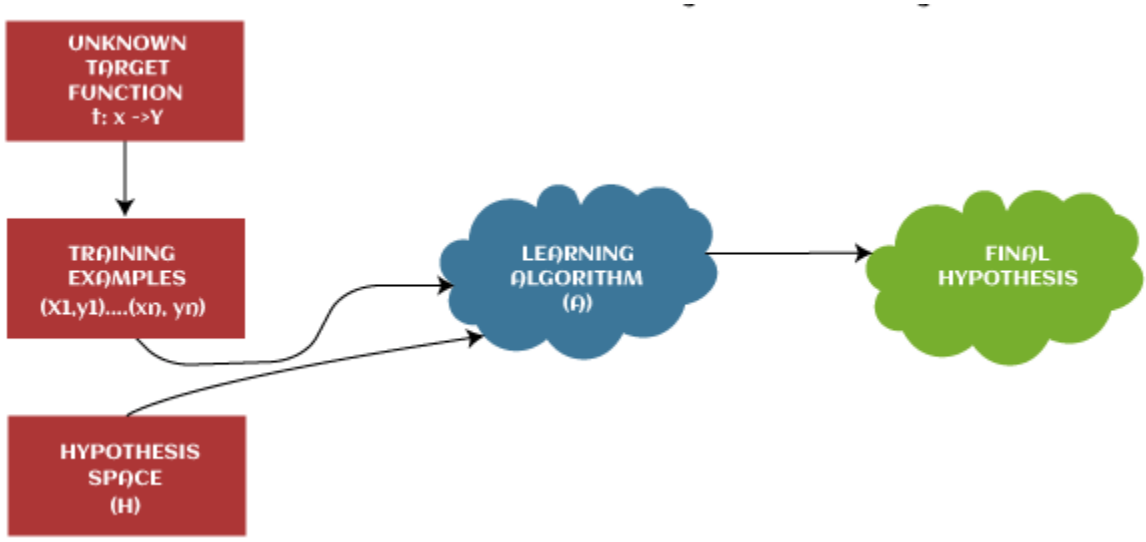
The hypothesis is defined as the supposition or proposed explanation based on insufficient evidence or assumptions. It is just a guess based on some known facts but has not yet been proven. A good hypothesis is testable, which results in either true or false.

Example: Let's understand the hypothesis with a common example. Some scientist claims that ultraviolet (UV) light can damage the eyes then it may also cause blindness.

In this example, a scientist just claims that UV rays are harmful to the eyes, but we assume they may cause blindness. However, it may or may not be possible. Hence, these types of assumptions are called a hypothesis.

Hypothesis in Machine Learning (ML)

The hypothesis is one of the commonly used concepts of statistics in Machine Learning. It is specifically used in Supervised Machine learning, where an ML model learns a function that best maps the input to corresponding outputs with the help of an available dataset.



In supervised learning techniques, the main aim is to determine the possible hypothesis out of hypothesis space that best maps input to the corresponding or correct outputs.

There are some common methods given to find out the possible hypothesis from the Hypothesis space, where hypothesis space is represented by **uppercase-h (H)** and hypothesis by **lowercase-h (h).** Th ese are defined as follows:

Hypothesis space (H):

Hypothesis space is defined as a set of all possible legal hypotheses; hence it is also known as a hypothesis set. It is used by supervised machine learning algorithms to determine the best possible hypothesis to describe the target function or best maps input to output.

It is often constrained by choice of the framing of the problem, the choice of model, and the choice of model configuration.

Hypothesis (h):

It is defined as the approximate function that best describes the target in supervised machine learning algorithms. It is primarily based on data as well as bias and restrictions applied to data.

Hence hypothesis (h) can be concluded as a single hypothesis that maps input to proper output and can be evaluated as well as used to make predictions.

The hypothesis (h) can be formulated in machine learning as follows:

y= mx + b

Where,

Y: Range

m: Slope of the line which divided test data or changes in y divided by change in x.

x: domain

c: intercept (constant)

Hypothesis Evaluation:

The process of machine learning involves not only formulating hypotheses but also evaluating their performance. This evaluation is typically done using a loss function or an evaluation metric that quantifies the disparity between predicted outputs and ground truth labels. Common evaluation metrics include mean squared error (MSE), accuracy, precision, recall, F1-score, and others. By comparing the predictions of the hypothesis with the actual outcomes on a validation or test dataset, one can assess the effectiveness of the model.

Sampling in machine learning:

It is the process of selecting a subset of data from a larger dataset. It's a fundamental step in the machine learning pipeline, and can be used for many purposes, including:

- **Reducing computational cost**

Sampling makes model training more efficient by working with a smaller, more manageable subset of data.

- **Handling imbalanced data**

Sampling can help ensure that diverse examples are included in the training data, which can lead to more robust models.

- **Preventing adversarial attacks**

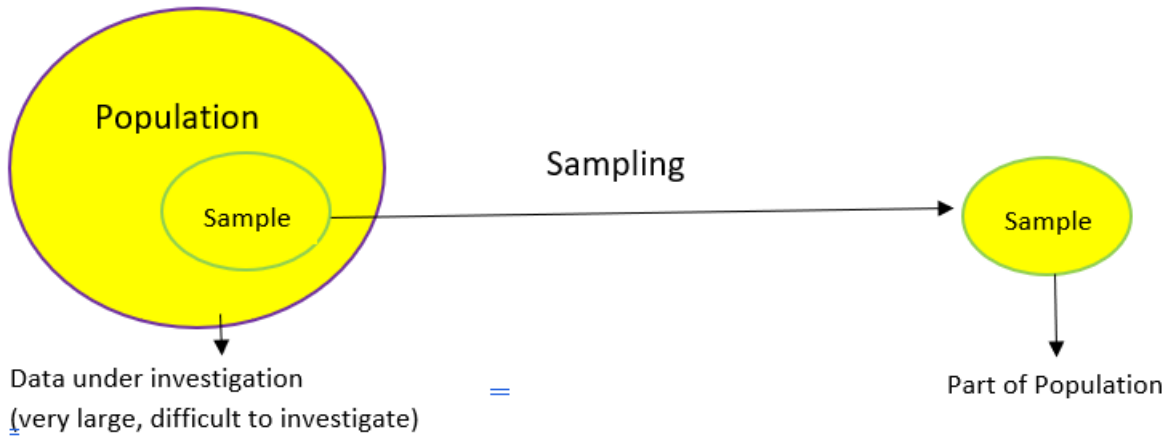
Sampling can help remove outliers, noise, or redundant data that can be exploited by adversarial attacks.

- **Exploratory data analysis**

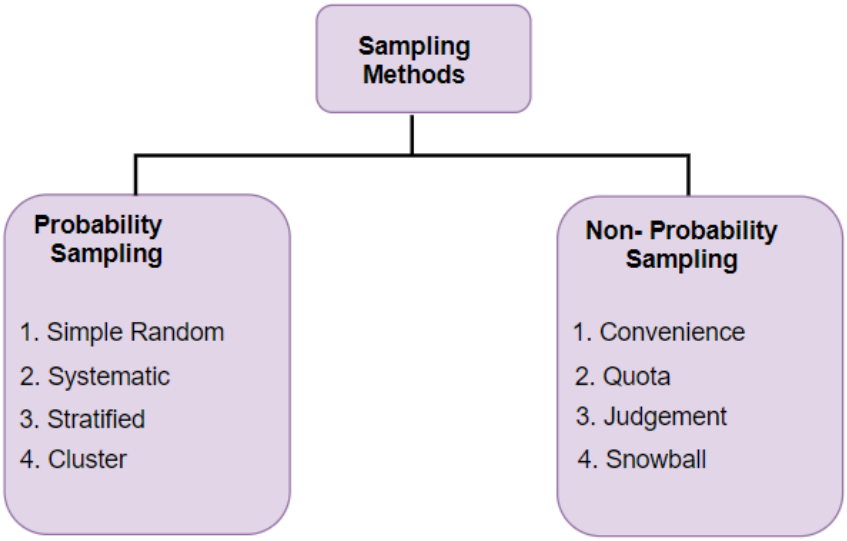
Sampling can provide manageable subsets for visualization and statistical analysis.

- **Working with anonymized data**

In industries where data privacy is critical, sampling can help data scientists work with anonymized or reduced subsets of data.



Different Types of Sampling



Here comes another diagrammatic illustration! This one talks about the different types of sampling techniques available to us:

- **Probability Sampling:** In probability sampling, every element of the population has an equal chance of being selected. Probability sampling gives us the best chance to create a sample that is truly representative of the population
- **Non-Probability Sampling:** In non-probability sampling, all elements do not have an equal chance of being selected. Consequently, there is a significant risk of ending up with a non-representative sample which does not produce generalizable results

Here are some commonly used sampling techniques:

Simple random sampling

- A straightforward approach where each data point has an equal probability of being selected. This technique minimizes bias and is suitable when the population is homogeneous. For example, a market research firm might use simple random sampling to gather customer feedback by randomly selecting respondents from their customer database.
- **Systematic sampling**

  Involves selecting data points at regular intervals from an ordered list or dataset. This technique is efficient and convenient, particularly when working with large datasets. It is commonly used in quality control processes, where manufacturers might inspect every nth product coming off the assembly line.

  Stratified sampling
- Divides the population into non-overlapping subgroups (strata) based on shared characteristics, like age, gender, or income level. A random sample is then taken from each subgroup, ensuring the sample accurately represents the population's diversity. In the picture above, we can see basketball players grouped by their position for a more representaive sample. This technique is useful in opinion polls and market research studies, where stratified sampling based on demographics can provide more reliable results.

  Cluster sampling
- Groups the population into clusters (e.g., households, neighborhoods, or schools), and a random sample of clusters is selected. All data points within the chosen clusters are included in the sample. This technique is practical when the population spreads over a large geographical area, such as in census data collection.

  Convenience sampling

- Involves selecting data points that are readily available or easily accessible. While convenient, this technique may not accurately represent the population. However, it can be useful for exploratory research or pilot studies when resources are limited.
- By selecting the appropriate technique, researchers can ensure their sample is representative, reliable, and aligned with their research objectives. This enables them to draw valid conclusions and insights from the data.

Quota Sampling

In the **Quota Sampling Method** of collecting data, the **entire population is divided into different classes or groups**. It is done on the **basis of the different characteristics of the given population**. The investigator fixes some percentages of the different groups with different characteristics of the total population. After that, he fixes some quota of the items for each of the selected segregated groups. At last, to form a sample, the investigator has to select a fixed number of items from each of the segregated groups.

Judgment Sampling

It is also known as selective sampling. It depends on the judgment of the experts when choosing whom to ask to participate.

Types of Machine Learning Algorithms

Machine Learning Algorithm can be broadly classified into three types:

1. **Supervised Learning Algorithms**
2. **Unsupervised Learning Algorithms**
3. **Reinforcement Learning algorithm**

The below diagram illustrates the different ML algorithm, along with the categories:

1) Supervised Learning Algorithm

Supervised learning is a type of Machine learning in which the machine needs external supervision to learn. The supervised learning models are trained using the labeled dataset. Once the training and processing are done, the model is tested by providing a sample test data to check whether it predicts the correct output.

The goal of supervised learning is to map input data with the output data. Supervised learning is based on supervision, and it is the same as when a student learns things in the teacher's supervision. The example of supervised learning is **spam filtering**.

Supervised learning can be divided further into two categories of problem:

- Classification

- Regression

Examples of some popular supervised learning algorithms are Simple Linear regression, Decision Tree, Logistic Regression, KNN algorithm, etc. Read more..

2) Unsupervised Learning Algorithm

It is a type of machine learning in which the machine does not need any external supervision to learn from the data, hence called unsupervised learning. The unsupervised models can be trained using the unlabelled dataset that is not classified, nor categorized, and the algorithm needs to act on that data without any supervision. In unsupervised learning, the model doesn't have a predefined output, and it tries to find useful insights from the huge amount of data. These are used to solve the Association and Clustering problems**. Hence further, it can be classified into two types:**

- Clustering
- Association

Examples of some Unsupervised learning algorithms are **K-means Clustering, Apriori Algorithm, Eclat, etc.**

3) Reinforcement Learning

In Reinforcement learning, an agent interacts with its environment by producing actions, and learn with the help of feedback. The feedback is given to the agent in the form of rewards, such as for each good action, he gets a positive reward, and for each bad action, he gets a negative reward. There is no supervision provided to the agent. **Q-Learning algorithm** is used in reinforcement learning.

Comparing machine learning algorithms: why we do it?

Comparing machine learning algorithms is valuable on its own, but there are some not-so-obvious benefits of effectively comparing various experiments. Let's take a look at the goals of comparison:

- **Better performance:** the primary objective of model comparison and selection is to improve the performance of the machine learning software/solution. The objective is to narrow down the best algorithms that suit the data and the business requirements.
- **Longer lifetime:** high performance can be short-lived if the chosen model is overly dependent on the training data and fails to generalize to unseen data. So, it's also important to select a model that captures the underlying data patterns so that the predictions remain accurate over time with minimal need for re-training.
- **Easier re-training:** when we evaluate models and prepare them for comparison, we also record documentation (the best parameters, configurations, results, etc.). These details ease retraining the model if there is a failure, because we don't need to redo the previous analysis. As a result, we can retrace the decisions made during initial model selection and find the potential causes for the failure (making it easier to adjust the model based on past

experiences). As a result, retraining can begin immediately and proceed with greater efficiency.

- **Speedy production:** with the model details available at hand, it's easy to narrow down on models that can offer high processing speed and that use memory resources optimally. Also during production, configuring machine learning solutions requires setting key parameters, such as memory usage, processing speed, and response time, to ensure optimal performance and resource efficiency.. Having production-level data can be useful for easily aligning with the production engineers. Moreover, knowing the resource demands of different algorithms, it will also be easier to check their compliance and feasibility concerning the organization's allocated assets.

Parameters of machine learning algorithms and how to compare them

Let's dive right into analyzing and understanding how to compare the different characteristics of algorithms that can be used to sort and choose the best machine learning models. I divided the comparable parameters into two high-level categories:

- development-based,
- and production-based parameters.

Development-based parameters

Statistical tests

On a fundamental level, machine learning models are statistical equations that run at great speed on multiple data points to arrive at a conclusion. Therefore, conducting statistical tests on the algorithms is critical to set them right and also to understand if the model's equation is the right fit for the dataset at hand. Here's a handful of popular statistical tests that can be used to set the grounds for comparison:

- **Null hypothesis testing:** null hypothesis testing is used to determine if the differences in two data samples or metric performances are statistically significant—meaning they reflect a true effect rather than random noise or coincidence.
- **ANOVA (Analysis Of Variance):** ANOVA is a statistical method used to determine whether there are significant differences between the means of three or more groups. For example, ANOVA can help reveal if different teaching methods result in different student scores or if all methods have similar effects. It uses one or more categorical independent variables (e.g., teaching method) to analyze their impact on a continuous dependent variable (e.g., student scores). Unlike Linear Discriminant Analysis (LDA), which is a classification technique, ANOVA focuses on comparing the means of the groups to assess variation.
- **Chi-Square:** it's a statistical tool or test that assesses the likelihood of association or correlation between categorical variables by comparing the observed and expected frequencies in each category.

- **Student's t-test:** it compares the means of two samples from normal distributions when the standard deviation is unknown to determine if the differences are statistically significant.
- **Ten-fold cross-validation:** the 10-fold cross-validation compares the performance of each algorithm on different datasets that have been configured with the same random seed to maintain uniformity in testing. Next, a hypothesis test like the student's paired t-test should be deployed to validate if the differences in metrics between the two models are statistically significant.

Model features and objectives

To choose the best machine learning model for a given dataset, it's essential to consider the features or parameters of the model. The parameters and model objectives help to gauge the model's flexibility, assumptions, and learning style.

When comparing linear regression models, we can choose between different ways to measure their errors. Some models try to minimize Mean Squared Error (MSE), while others aim to reduce Mean Absolute Error (MAE). The choice really comes down to how we want to handle outliers in our data.

If we have outliers in our dataset and want to consider them without letting them skew our results, using MAE makes more sense. The reason is pretty straightforward: MAE just takes the absolute value of errors, so it treats all deviations more evenly. MSE, on the other hand, squares the errors, which makes extreme values have a much bigger impact on the final model. So when we want outliers to matter but not take over, an MAE-based model tends to work better.

Similarly for classification, if two models (for example, decision tree and random forest) are considered, then the primary basis for comparison will be the degree of generalization that the model can achieve. A decision tree model with just one tree will have a limited ability to reduce variance through the max_depth parameter, whereas a random forest model will have an extended ability to bring generalization via both max_depth and n_estimators parameters.

Bayes Theorem in Machine Learning

Bayes theorem is given by an English statistician, philosopher, and Presbyterian minister named **Mr. Thomas Bayes** in 17th century. Bayes provides their thoughts in decision theory which is extensively used in important mathematics concepts as Probability. Bayes theorem is also widely used in Machine Learning where we need to predict classes precisely and accurately. An important concept of Bayes theorem named **Bayesian method** is used to calculate conditional probability in Machine Learning application that includes classification tasks. Further, a simplified version of Bayes theorem (Naïve Bayes classification) is also used to reduce computation time and average cost of the projects.

Bayes theorem is also known with some other name such as **Bayes rule or Bayes Law.** ***Bayes theorem helps to determine the probability of an event with random knowledge***. It is used to

calculate the probability of occurring one event while other one already occurred. It is a best method to relate the condition probability and marginal probability.

In simple words, we can say that Bayes theorem helps to contribute more accurate results.

Bayes Theorem is used to estimate the precision of values and provides a method for calculating the conditional probability. However, it is hypocritically a simple calculation but it is used to easily calculate the conditional probability of events where intuition often fails. Some of the data scientist assumes that Bayes theorem is most widely used in financial industries but it is not like that. Other than financial, Bayes theorem is also extensively applied in health and medical, research and survey industry, aeronautical sector, etc.

What is Bayes Theorem?

Bayes theorem is one of the most popular machine learning concepts that helps to calculate the probability of occurring one event with uncertain knowledge while other one has already occurred.

Bayes' theorem can be derived using product rule and conditional probability of event X with known event Y:

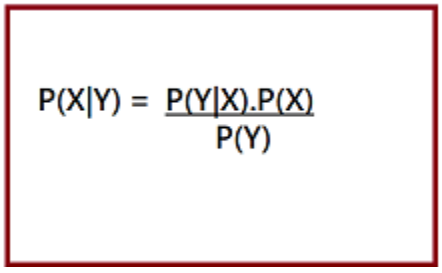
- According to the product rule we can express as the probability of event X with known event Y as follows;

1. P(X ? Y)= P(X|Y) P(Y) {equation 1}

- Further, the probability of event Y with known event X:

1. P(X ? Y)= P(Y|X) P(X) {equation 2}

Mathematically, Bayes theorem can be expressed by combining both equations on right hand side. We will get:



Here, both events X and Y are independent events which means probability of outcome of both events does not depends one another.

The above equation is called as Bayes Rule or Bayes Theorem.

- P(X|Y) is called as **posterior**, which we need to calculate. It is defined as updated probability after considering the evidence.
- P(Y|X) is called the likelihood. It is the probability of evidence when hypothesis is true.

- P(X) is called the **prior probability**, probability of hypothesis before considering the evidence
- P(Y) is called marginal probability. It is defined as the probability of evidence under any consideration.

Hence, Bayes Theorem can be written as:

posterior = likelihood * prior / evidence

Prerequisites for Bayes Theorem

While studying the Bayes theorem, we need to understand few important concepts. These are as follows:

1. Experiment

An experiment is defined as the planned operation carried out under controlled condition such as tossing a coin, drawing a card and rolling a dice, etc.

2. Sample Space

During an experiment what we get as a result is called as possible outcomes and the set of all possible outcome of an event is known as sample space. For example, if we are rolling a dice, sample space will be:

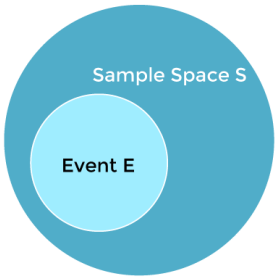
S1 = {1, 2, 3, 4, 5, 6}

Similarly, if our experiment is related to toss a coin and recording its outcomes, then sample space will be:

S2 = {Head, Tail}

3. Event

Event is defined as subset of sample space in an experiment. Further, it is also called as set of outcomes.



Assume in our experiment of rolling a dice, there are two event A and B such that;

A = Event when an even number is obtained = {2, 4, 6}

B = Event when a number is greater than 4 = {5, 6}

- **Probability of the event A "P(A)"**= Number of favourable outcomes / Total number of possible outcomes
  P(E) = 3/6 =1/2 =0.5
- Similarly, **Probability of the event B "P(B)"**= Number of favourable outcomes / Total number of possible outcomes
  =2/6
  =1/3
  =0.333

Bayes Optimal Classifier

The ***Bayes Optimal Classifier*** is a theoretical model that provides the most accurate classification of a new instance based on the training data. It operates under the principles of Bayes' theorem, calculating the conditional probabilities of different outcomes and selecting the one with the highest probability. This classifier is often referred to as the Bayes optimal learner, and it serves as a benchmark for evaluating the performance of other classifiers in machine learning.

Key Concepts

1. **Bayes' Theorem**: At the core of the Bayes Optimal Classifier is Bayes' theorem, which describes how to update the probability of a hypothesis based on new evidence. The theorem is expressed mathematically as:

$$P(H|E)=P(E|H)\cdot P(H)P(E)$$

Where:

- $P(H|E)P(H|E)$ is the posterior probability of the hypothesis HH given evidence EE.
- $P(E|H)P(E|H)$ is the likelihood of observing evidence EE given hypothesis HH.
- $P(H)P(H)$ is the prior probability of hypothesis HH.
- $P(E)P(E)$ is the marginal likelihood of evidence EE.

2. **Maximum A Posteriori (MAP)**: This is a probabilistic framework that seeks to find the most probable hypothesis given the training data. It is closely related to the Bayes Optimal Classifier but focuses on selecting a single hypothesis rather than making a prediction based on all possible hypotheses.
3. **Hypothesis Space**: The set of all possible hypotheses that can be used to classify the data. The Bayes Optimal Classifier evaluates each hypothesis and combines their predictions based on their posterior probabilities.

Naïve Bayes Classifier Algorithm

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems.

- It is mainly used in *text classification* that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- **It is a probabilistic classifier, which means it predicts on the basis of the probability of an object**.
- Some popular examples of Naïve Bayes Algorithm are **spam filtration, Sentimental analysis, and classifying articles**.

Why is it called Naïve Bayes?

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

- **Naïve**: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- **Bayes**: It is called Bayes because it depends on the principle of Bayes' Theorem.

Working of Naïve Bayes' Classifier:

Working of Naïve Bayes' Classifier can be understood with the help of the below example:

Suppose we have a dataset of **weather conditions** and corresponding target variable "**Play**". So using this dataset we need to decide that whether we should play or not on a particular day according to the weather conditions. So to solve this problem, we need to follow the below steps:

1. Convert the given dataset into frequency tables.
2. Generate Likelihood table by finding the probabilities of given features.
3. Now, use Bayes theorem to calculate the posterior probability.

Advantages of Naïve Bayes Classifier:

- Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
- It can be used for Binary as well as Multi-class Classifications.
- It performs well in Multi-class predictions as compared to the other Algorithms.
- It is the most popular choice for **text classification problems**.

Disadvantages of Naïve Bayes Classifier:

- Naive Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.

Applications of Naïve Bayes Classifier:

- It is used for **Credit Scoring**.
- It is used in **medical data classification**.
- It can be used in **real-time predictions** because Naïve Bayes Classifier is an eager learner.
- It is used in Text classification such as **Spam filtering** and **Sentiment analysis**.

Types of Naïve Bayes Model:

There are three types of Naive Bayes Model, which are given below:

- **Gaussian**: The Gaussian model assumes that features follow a normal distribution. This means if predictors take continuous values instead of discrete, then the model assumes that these values are sampled from the Gaussian distribution.
- **Multinomial**: The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed. It is primarily used for document classification problems, it means a particular document belongs to which category such as Sports, Politics, education, etc. The classifier uses the frequency of words for the predictors.
- **Bernoulli**: The Bernoulli classifier works similar to the Multinomial classifier, but the predictor variables are the independent Booleans variables. Such as if a particular word is present or not in a document. This model is also famous for document classification tasks.

What is an EM algorithm?

The Expectation-Maximization (EM) algorithm is defined as the combination of various unsupervised machine learning algorithms, which is used to determine the **local maximum likelihood estimates (MLE)** or **maximum a posteriori estimates (MAP)** for unobservable variables in statistical models. Further, it is a technique to find maximum likelihood estimation when the latent variables are present. It is also referred to as the **latent variable model.**

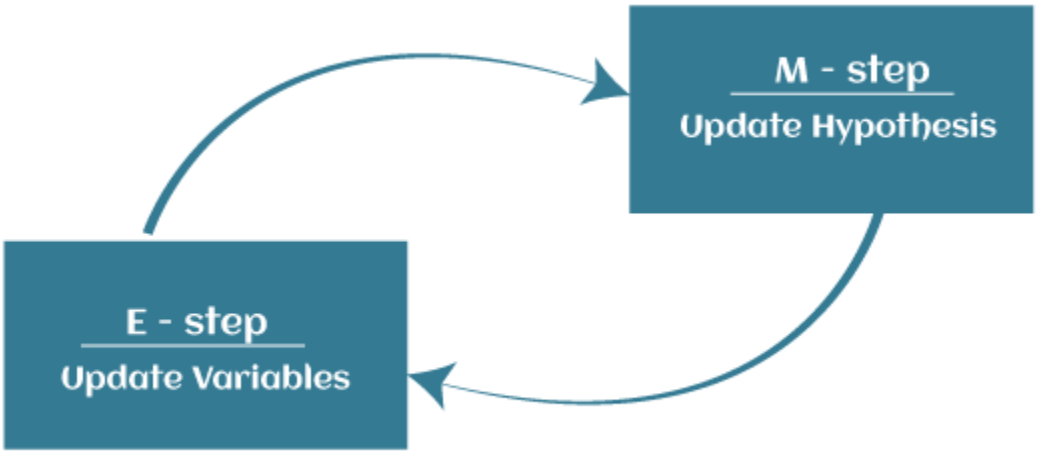
A latent variable model consists of both observable and unobservable variables where observable can be predicted while unobserved are inferred from the observed variable. These unobservable variables are known as latent variables.

Key Points:

- It is known as the latent variable model to determine MLE and MAP parameters for latent variables.
- It is used to predict values of parameters in instances where data is missing or unobservable for learning, and this is done until convergence of the values occurs.

EM Algorithm

The EM algorithm is the combination of various unsupervised ML algorithms, such as the **k-means clustering algorithm**. Being an iterative approach, it consists of two modes. In the first mode, we estimate the missing or latent variables. Hence it is referred to as the **Expectation/estimation step (E-step**). Further, the other mode is used to optimize the parameters of the models so that it can explain the data more clearly. The second mode is known as the **maximization-step or M-step.**



- **Expectation step (E - step):** It involves the estimation (guess) of all missing values in the dataset so that after completing this step, there should not be any missing value.
- **Maximization step (M - step):** This step involves the use of estimated data in the E-step and updating the parameters.
- **Repeat E-**step and M-step until the convergence of the values occurs.

The primary goal of the EM algorithm is to use the available observed data of the dataset to estimate the missing data of the latent variables and then use that data to update the values of the parameters in the M-step.

What is Convergence in the EM algorithm?

Convergence is defined as the specific situation in probability based on intuition, e.g., if there are two random variables that have very less difference in their probability, then they are known as converged. In other words, whenever the values of given variables are matched with each other, it is called convergence.

Steps in EM Algorithm

The EM algorithm is completed mainly in 4 steps, which include I***nitialization Step, Expectation Step, Maximization Step, and convergence Step***. These steps are explained as follows:

- **1st Step:** The very first step is to initialize the parameter values. Further, the system is provided with incomplete observed data with the assumption that data is obtained from a specific model.
- **2nd Step:** This step is known as Expectation or E-Step, which is used to estimate or guess the values of the missing or incomplete data using the observed data. Further, E-step primarily updates the variables.
- **3rd Step:** This step is known as Maximization or M-step, where we use complete data obtained from the 2nd step to update the parameter values. Further, M-step primarily updates the hypothesis.
- **4th step:** The last step is to check if the values of latent variables are converging or not. If it gets "yes", then stop the process; else, repeat the process from step 2 until the convergence occurs.

Applications of EM algorithm

The primary aim of the EM algorithm is to estimate the missing data in the latent variables through observed data in datasets. The EM algorithm or latent variable model has a broad range of real-life applications in machine learning. These are as follows:

- The EM algorithm is applicable in data clustering in machine learning.
- It is often used in computer vision and NLP (Natural language processing).

- It is used to estimate the value of the parameter in mixed models such as the **Gaussian Mixture Model**and quantitative genetics.
- It is also used in psychometrics for estimating item parameters and latent abilities of item response theory models.
- It is also applicable in the medical and healthcare industry, such as in image reconstruction and structural engineering.
- It is used to determine the Gaussian density of a function.

Advantages of EM algorithm

- It is very easy to implement the first two basic steps of the EM algorithm in various machine learning problems, which are E-step and M- step.
- It is mostly guaranteed that likelihood will enhance after each iteration.
- It often generates a solution for the M-step in the closed form.

Disadvantages of EM algorithm

- The convergence of the EM algorithm is very slow.
- It can make convergence for the local optima only.
- It takes both forward and backward probability into consideration. It is opposite to that of numerical optimization, which takes only forward probabilities.