**Naïve Bayes Classifier**

Agenda

* Understand the mathematical logic of the algorithm
* Figure out the parameters and hyper-parameters in the algorithm
* Mathematical importance and logic behind each of those parameters and hyper-parameters.
* Applications of the algorithm.

1.Understand the mathematical logic of the algorithm .

* **Understanding Naive Bayes and Machine Learning**

Machine learning falls into [two categories](https://www.simplilearn.com/tutorials/machine-learning-tutorial/types-of-machine-learning):

* Supervised Learning
* UnSupervised Learning

Supervised Learning falls into two categories:

* Classification
* Regression

Naive Bayes algorithm falls under classification.

* Naïve Bayes – I*ntroduction*
* Classification technique based on Bayes ‘ Theorem
* With “naïve” assumption of independence among predictors.
* Easy to build
* Particularly useful for very large data sets
* Known to outperform even highly sophisticated classification methods
* E.g. Earlier method for spam detection
* Naïve Bayes Algorithm
* A simple classifier that performs surprisingly well on a large class of problem.
* This type of methods are called Generative Learning Models
* Where is Naïve Bayes Used?

You can use Naive Bayes for the following things:

* Face Recognition
* Whether Prediction
* Medical Diagnosis
* News Classification
* **Understanding Naïve Bayes Classifier**
* Based on the Bayes theorem, the Naive Bayes Classifier gives the conditional probability of an event A given event B.
* Introduction – Bayes theorem
* P(c|x) – the posterior probability of class(c, target) given predictor (x, attributes).
* P(c) – the prior probability of class
* P(x|c ) – is the likelihood which of the probability of predictor given class

****

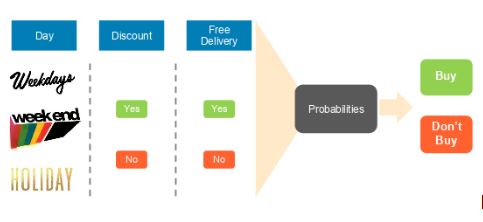
* Working of Naïve Bayes Classifier

1. Convert the Given dataset into Frequency tables.
2. Generate likelihood table by finding the probabilities of given feature.
3. Now, use Bayes theorem to Calculate the posterior probability.

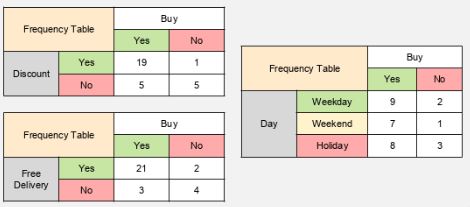
Let us use the following demo to understand the concept of a Naive Bayes classifier:

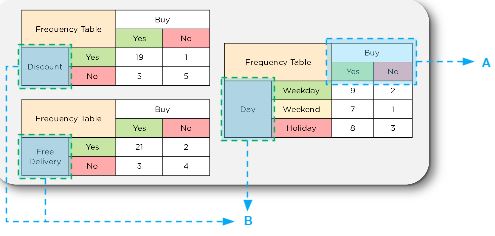
**Shopping Example**

* Problem statement: To predict whether a person will purchase a product on a specific combination of day, discount, and free delivery using a Naive Bayes classifier.

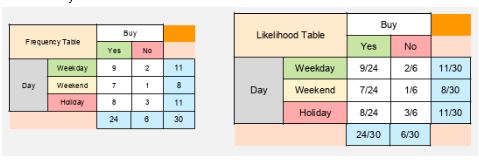


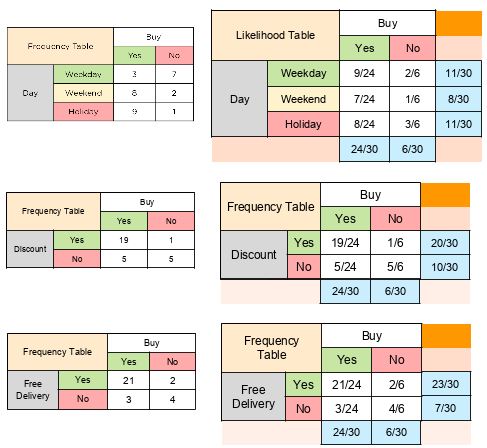
Based on the dataset containing the three input types—day, discount, and free delivery— the frequency table for each attribute is populated.



For Bayes theorem, let the event ‘buy’ be A and the independent variables (discount, free delivery, day) be B. 

Let us calculate the likelihood for one of the “day” variables, which includes weekday, weekend, and holiday variables.

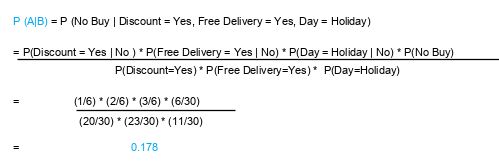




The likelihood tables can be used to calculate whether a customer will purchase a product on a specific combination of the day when there is a discount and whether there is free delivery. Consider a combination of the following factors where B equals:

* Day = Holiday
* Discount = Yes
* Free Delivery = Yes
* Let us find the probability of them not purchasing based on the conditions above.
* A = No Purchase

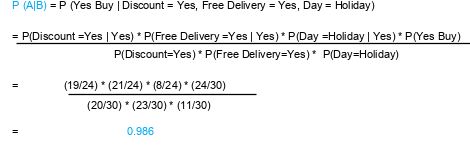
Applying Bayes Theorem, we get P(A | B) as shown:



Similarly, let us find the probability of them purchasing a product under the conditions above.

Here, A = Buy

Applying Bayes Theorem, we get P(A | B) as shown:



From the two calculations above, we find that:

Probability of purchase = 0.986

Probability of no purchase = 0.178

Finally, we have a conditional probability of purchase on this day.

Next, normalize these probabilities to get the likelihood of the events:

Sum of probabilities = 0.986 + 0.178 = 1.164

Likelihood of purchase = 0.986 / 1.164 = 84.71 percent

Likelihood of no purchase = 0.178 / 1.164 = 15.29 percent

Result: As 84.71 percent is greater than 15.29 percent, we can conclude that an average

customer will buy on holiday with a discount and free delivery.

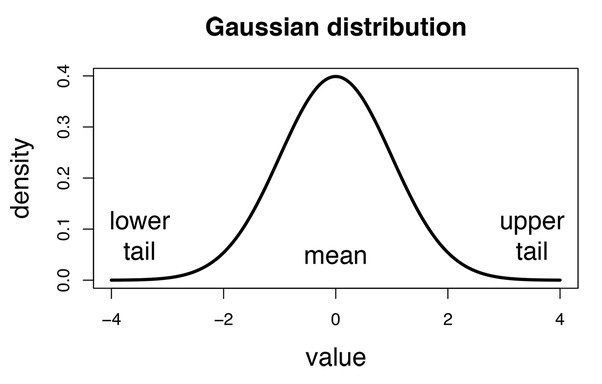
After understanding how Naive Bayes Classifier works, we can explore its benefits.

**2.** Figure out the parameters and hyper-parameters in the algorithm

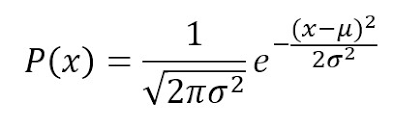
* Types of Naïve Bayes Classifier
* Gaussian Naïve Bayes
* Multinomial Naïve Bayes
* Bernoulli Naïve Bayes
* Categorical Naïve Bayes

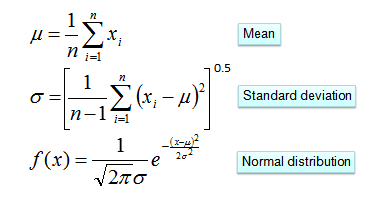
Gaussian Naïve Bayes

In Gaussian Naïve Bayes, continuous values associated with each feature are assumed to be distributed according to a **Gaussian distribution (**[Normal distribution](https://en.wikipedia.org/wiki/Normal_distribution)**)**. When plotted, it gives a bell-shaped curve which is symmetric about the mean of the feature values as shown below:



The likelihood of the features is assumed to be Gaussian, hence, conditional probability is given by:



The probability density function for the normal distribution is defined by two parameters (mean and standard deviation)

*class*sklearn.naive\_bayes.**GaussianNB**(*\**, *priors=None*, *var\_smoothing=1e-09*)

**Parameters:**

**Priors : *array-like of shape (n\_classes), default=None***

Prior probabilities of the classes. If specified, the priors are not adjusted according to the data.

**var\_smoothing : *float, default=1e-9***

Portion of the largest variance of all features that is added to variances for calculation stability.

Multinomial Naïve Bayes:

This is mostly used for document classification problem, i.e whether a document belongs to the category of sports, politics, technology etc. The features/predictors used by the classifier are the frequency of the words present in the document.

*class*sklearn.naive\_bayes.**MultinomialNB**(*\**, *alpha=1.0*, *force\_alpha='warn'*, *fit\_prior=True*, *class\_prior=None*)

**Parameters:**

**Alpha : *float or array-like of shape (n\_features,), default=1.0***

Additive (Laplace/Lidstone) smoothing parameter (set alpha=0 and force\_alpha=True, for no smoothing).

**force\_alpha : *bool, default=False***

If False and alpha is less than 1e-10, it will set alpha to 1e-10. If True, alpha will remain unchanged. This may cause numerical errors if alpha is too close to 0.

**fit\_prior : *bool, default=True***

Whether to learn class prior probabilities or not. If false, a uniform prior will be used.

**class\_prior*array-like of shape (n\_classes,), default=None***

Prior probabilities of the classes. If specified, the priors are not adjusted according to the data.

## Bernoulli Naive Bayes: This is similar to the multinomial naive bayes but the predictors are boolean variables. The parameters that we use to predict the class variable take up only values yes or no, for example if a word occurs in the text or not.

*class*sklearn.naive\_bayes.**BernoulliNB**(*\**, *alpha=1.0*, *force\_alpha='warn'*, *binarize=0.0*, *fit\_prior=True*, *class\_prior=None*)

**Parameters:**

**Alpha : *float or array-like of shape (n\_features,), default=1.0***

Additive (Laplace/Lidstone) smoothing parameter (set alpha=0 and force\_alpha=True, for no smoothing).

**force\_alpha : *bool, default=False***

If False and alpha is less than 1e-10, it will set alpha to 1e-10. If True, alpha will remain unchanged. This may cause numerical errors if alpha is too close to 0.

**Binarize : *float or None, default=0.0***

Threshold for binarizing (mapping to booleans) of sample features. If None, input is presumed to already consist of binary vectors.

**fit\_prior : *bool, default=True***

Whether to learn class prior probabilities or not. If false, a uniform prior will be used.

Categorical Naïve Bayes: The categorical Naive Bayes classifier is suitable for classification with discrete features that are categorically distributed. The categories of each feature are drawn from a categorical distribution.

*class*sklearn.naive\_bayes.**CategoricalNB**(*\**, *alpha=1.0*, *force\_alpha='warn'*, *fit\_prior=True*, *class\_prior=None*, *min\_categories=None*

**Parameters:**

**Alpha : float, default=1.0**

Additive (Laplace/Lidstone) smoothing parameter (set alpha=0 and force\_alpha=True, for no smoothing).

**force\_alpha : *bool, default=False***

If False and alpha is less than 1e-10, it will set alpha to 1e-10. If True, alpha will remain unchanged. This may cause numerical errors if alpha is too close to 0.

**fit\_prior : *bool, default=True*** Whether to learn class prior probabilities or not. If false, a uniform prior will be used.

**class\_prior : *array-like of shape (n\_classes,), default=None***

Prior probabilities of the classes. If specified, the priors are not adjusted according to the data.

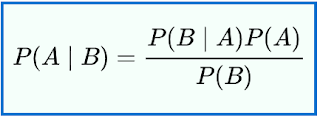
**min\_categories : *int or array-like of shape (n\_features,), default=None***

Minimum number of categories per feature.

* integer: Sets the minimum number of categories per feature to n\_categories for each features.
* array-like: shape (n\_features,) where n\_categories[i] holds the minimum number of categories for the ith column of the input.
* None (default): Determines the number of categories automatically from the training data.

3. Mathematical importance and logic behind each of those parameters and hyper-parameters.

Bayes Formula:



* How did we get this?

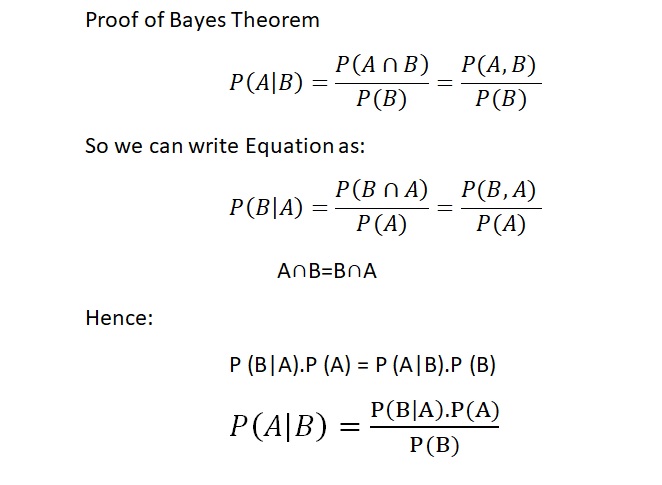
Bayes Theorem – Statement

Bayes' theorem is stated mathematically as the following equation



where � and �are [events](https://en.wikipedia.org/wiki/Event_(probability_theory)) and P(B) 0�(�)≠0PP

* �(�∣�)P(A|B) is Posterior Probability: Probability of hypothesis A on the observed event B�.
* �(�∣�)P(B|A) is Likelihood Probability: Probability of the evidence given that the probability of a hypothesis is true.
* �(�)P(A) is Prior Probability: Probability of hypothesis before observing the evidence.
* P(B)P(B) is Marginal Probability: Probability of Evidence



\begin{eqnarray*}
P(y \vert f_1, \ldots, f_m) &=& \frac{P(f_1, \ldots, f_m \...
...
&=& \textmd{arg max}_{y} P(y) \prod_{i = 1}^m P(f_i \vert y)
\end{eqnarray*}

Parameter and Hyper parameter naïve Bayes classifier

The parameters that are learned in Naive Bayes are the *prior probabilities* of different classes, as well as the *likelihood* of different features for each class. In the test phase, these learned parameters are used to estimate the probability of each class for the given sample.

In [Bayesian statistics](https://en.wikipedia.org/wiki/Bayesian_statistics), a **hyperparameter** is a parameter of a [prior distribution](https://en.wikipedia.org/wiki/Prior_distribution); the term is used to distinguish them from parameters of the model for the underlying system under analysis.

For example, if one is using a [beta distribution](https://en.wikipedia.org/wiki/Beta_distribution) to model the distribution of the parameter *p* of a [Bernoulli distribution](https://en.wikipedia.org/wiki/Bernoulli_distribution), then:

* *p* is a parameter of the underlying system (Bernoulli distribution), and
* *α* and *β* are parameters of the prior distribution (beta distribution), hence *hyper*parameters.

One may take a single value for a given hyperparameter, or one can iterate and take a probability distribution on the hyperparameter itself, called a [hyperprior](https://en.wikipedia.org/wiki/Hyperprior).

4. **Applications of Naive Bayes Algorithms**

* **Real-time Prediction:**Naive Bayesian classifier is an eager learning classifier and it is super fast. Thus, it could be used for making predictions in real time.
* **Multi-class Prediction:**This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.
* **Text classification/ Spam Filtering/ Sentiment Analysis:** Naive Bayesian classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)
* **Recommendation System:**Naive Bayes Classifier and [Collaborative Filtering](https://en.wikipedia.org/wiki/Collaborative_filtering) together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not.
* Advantages of Naïve Bayes Classifier
* It is simple and easy to implement.
* It doesn't require as much training data.
* It handles both continuous and discrete data.
* It is highly scalable with the number of predictors and data points.
* It is fast and can be used to make real-time predictions.
* Disadvantages of Naïve Bayes Classifier
* **Zero probability problem :**When we encounter words in the test data for a particular class that are not present in the training data, we might end up with zero class probabilities.

Reference

<https://www.simplilearn.com/tutorials/machine-learning-tutorial/naive-bayes-classifier#understanding_naive_bayes_and_machine_learning>

<https://towardsdatascience.com/introduction-to-na%C3%AFve-bayes-classifier-fa59e3e24aaf>

<https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.CategoricalNB.html#sklearn.naive_bayes.CategoricalNB>

<https://www.youtube.com/watch?v=jS1CKhALUBQ&list=PLZoTAELRMXVPBTrWtJkn3wWQxZkmTXGwe&index=81>