Table of Contents

[**I.** **Naïve Bayes Classifiers** 2](#_Toc127559514)

[1. Why is it call Naïve Bayes? 2](#_Toc127559515)

[2. Bayes’ Theorem 3](#_Toc127559516)

[3. Working of Naïve Bayes’ Classifier: 4](#_Toc127559517)

[4. Advantages of Naïve Bayes Classifier: 5](#_Toc127559518)

[5. Disadvantages of Naïve Bayes Classifier: 5](#_Toc127559519)

[6. Applications of Naïve Bayes Classifier: 6](#_Toc127559520)

[7. Naïve Bayes Models: 6](#_Toc127559521)

[*a)* *Gaussian Naïve Bayes* 6](#_Toc127559522)

[*b)* *Multinomial Naïve Bayes* 6](#_Toc127559523)

[*c)* *Bernoulli Naïve Bayes* 6](#_Toc127559524)

[*d)* *Complement Naïve Bayes* 7](#_Toc127559525)

[*e)* *Categorical Naïve Bayes* 7](#_Toc127559526)

[**II.** **Python Implementation** 7](#_Toc127559527)

[**III.** **Reference** 7](#_Toc127559528)

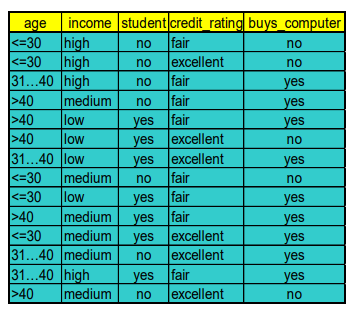
# **Naïve Bayes Classifiers**

* Naïve Bayes classifiers are a collection of supervised learning algorithms based on Bayes’ theorem.
* It is a family of algorithms having the same principle: every pair of features being classified is independent of each other.
* This method includes:
  + Gaussian Naïve Bayes
  + Multinomial Naïve Bayes
  + Complement Naïve Bayes
  + Bernoulli Naïve Bayes
  + Categorical Naïve Bayes

## Why is it call Naïve Bayes?

* **Naïve**: this method assumes that the occurrence of a feature is independent of the occurrence of the others. Plus, their precedents are the same to contribution to the outcome.

To illustrate, give the table below:



The “age” feature, which is lower than or equal to 30 or it is higher than 40, does not affect to how the “income” is hight or low and so on.

Lack one of 4 features will not determine the “buys\_computer” is whether no or yes.

* **Bayes**: the principle of Bayes’ Theorem.

## Bayes’ Theorem

Bayes’ Theorem finds the probability of a hypothesis with prior knowledge. In other words, that is the probability of an event which occurs when the probability of another event has already occurred.

The formula:

Where:

* + **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) is Marginal Probability**: Probability of Evidence.

**Regarding to the dataset, we rewrite as below:**

Where:

* y is class variable
* X is a dependent feature vector.

X = (x1, x2, …, xn)

For example:

* y = (buy\_computer)
* X = (age, income, student, credit\_rating)

The denominator is constant, therefore:

We can use Maximum A Posteriori (MAP) estimation to estimate P(y), the relative frequency of class y in the training set, and P(xi|y).

The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of P(xi|y).

## Working of Naïve Bayes’ Classifier:

* + Step 1: Convert the dataset into frequency tables.
  + Step 2: Generate Likelihood table by finding the probabilities of given features.
  + Step 3: Use Bayes to calculate the posterior probability.

Give dataset:

|  |  |  |
| --- | --- | --- |
|  | Outlook | Play |
| 0 | Rainy | Yes |
| 1 | Sunny | Yes |
| 2 | Overcast | Yes |
| 3 | Overcast | Yes |
| 4 | Sunny | No |
| 5 | Rainy | Yes |
| 6 | Sunny | Yes |
| 7 | Overcast | Yes |
| 8 | Rainy | No |
| 9 | Sunny | No |
| 10 | Sunny | Yes |
| 11 | Rainy | No |
| 12 | Overcast | Yes |
| 13 | Overcast | Yes |

Frequency table:

|  |  |  |
| --- | --- | --- |
| Weather | Yes | No |
| Overcast | 5 | 0 |
| Rainy | 2 | 2 |
| Sunny | 3 | 2 |
| Total | 10 | 5 |

Likelihood table:

|  |  |  |  |
| --- | --- | --- | --- |
| Weather | No | Yes |  |
| Overcast | 0 | 5 | 5/14 |
| Rainy | 2 | 2 | 4/14 |
| Sunny | 2 | 3 | 5/14 |
| All | 4/14 | 10/14 |  |

**Applying Bayes' theorem:**

**P(Yes|Sunny) = P(Sunny| Yes)\*P(Yes)/P(Sunny)**

P(Sunny|Yes) = 3/10= 0.3

P(Sunny) = 0.35

P(Yes) = 0.71

* P(Yes|Sunny) = 0.3\*0.71/0.35= **0.60**

**P(No|Sunny) = P(Sunny| No)P(No)/P(Sunny)**

P(Sunny|NO) = 2/4=0.5

P(No)= 0.29

P(Sunny)= 0.35

* P(No|Sunny) = 0.5\*0.29/0.35 = **0.41**
* **P(Yes|Sunny)>P(No|Sunny)**

**Hence on a Sunny day, Player can play the game.**

## 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-classes 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**.

## **Naïve Bayes Models:**

### *Gaussian Naïve Bayes*

GaussianNB implements the Gaussian Naive Bayes algorithm for classification:

### *Multinomial Naïve Bayes*

MultinomialNB is used for multinomially distributed data and document classification problems, which is represented as word vector counts.

Where:

* is the number of times feature i appears in a sample of class y in the training set T
* is the total count of all features for class y
* The smoothing prior accounts for features not present in the learning samples and prevents zero probabilities in further computations
  + : Laplace smoothing
  + : Lidstone smoothing

### *Bernoulli Naïve Bayes*

Famous for document classification tasks, BernoulliNB is used for the data having multiple features but each one is assumed to be a binary-valued (Bernoulli, Boolean) variable. Such as if a particular word is present or not in a document.

### *Complement Naïve Bayes*

<https://scikit-learn.org/stable/modules/naive_bayes.html#bernoulli-naive-bayes>

### *Categorical Naïve Bayes*

<https://scikit-learn.org/stable/modules/naive_bayes.html#categorical-naive-bayes>

# **Python Implementation**

Steps to implement:

* Data pre-processing step
* Fitting Naïve Bayes to the Training set
* Predicting the test result
* Test accuracy of the result
* Visualizing the test set result

My notebook: [click me!](https://colab.research.google.com/drive/1Va_zqDOtDecNyTP7Ci3o22QkigfBDpT2?usp=sharing)

# **Reference**

[1.9. Naive Bayes — scikit-learn 1.2.1 documentation](https://scikit-learn.org/stable/modules/naive_bayes.html#bernoulli-naive-bayes)

[Naive Bayes Classifier in Machine Learning - Javatpoint](https://www.javatpoint.com/machine-learning-naive-bayes-classifier)

[Naive Bayes Classifiers - GeeksforGeeks](https://www.geeksforgeeks.org/naive-bayes-classifiers/)