Machine Learning to build Intelligent Systems

Manas Dasgupta

Naïve Bayes Classifier

Structure of this Module

Naïve Bayes Classifier

TOPICS
Introduction to Naïve Bayes
Bayes Theorem
Naïve Bayes Assumption
Gaussian Naïve Bayes
Multinomial Naïve Bayes
Naïve Bayes Calculation
Python Demo

Introduction to Naïve Bayes

The ***Naïve Bayes classifier*** is a simple probabilistic classifier which is based on **Bayes theorem** but with strong **assumptions regarding independence**. Historically, this technique became popular with applications in email filtering, spam detection, and document categorization.

Although it is often outperformed by other techniques, and despite the naïve design and oversimplified assumptions, this classifier can perform well in many complex real-world problems. And since it is a resource efficient algorithm that is fast and scales well, it has become a very popular Machine Learning Algorithm.

Some cool use cases of Naïve Bayes

Use Case	Description
Spam Filtering	Email Classification as Spam or Ham
Sentiment Analysis	Using in conjunction with NLP techniques, Naïve Bayes can be used to assign Sentiment Scores to texts.
Disease Detection	Based on existing conditions diagnosed, a Naïve Bayes model can detect whether or not a person has a specific disease.
Weather Forecasting	Based on conditions existing today, a Naïve Bayes model can predict weather conditions for tomorrow.

Simple

Works well with high Dimensional Data

Fast & Scalable

Used often for Benchmarking

Based on Bayesian Classification

Introduction to Naïve Bayes

Naive Bayes classifiers are built on **Bayesian classification methods**. These rely on Bayes's theorem, which is an **equation describing the relationship of conditional probabilities of statistical quantities**.

In Bayesian classification, we're interested in ***finding the probability of a label given some observed features***, which we can write as ***P(L|Features)***. Bayes's theorem tells us how to express this in terms of quantities we can compute more directly:

$$P(L \mid \text{features}) = \frac{P(\text{features} \mid L)P(L)}{P(\text{features})}$$

Naïve Bayes Theorem

For example, we are to examine Employee Attrition Data with a goal of creating a Naïve Bayes Model that could predict Attritions. Here, we are seeking the probability of an employee belonging to attrition class $\boldsymbol{C_k}$ (where C_{yes}=attrition and $\mathbf{C}_{no}$=non-attrition) given that its predictor values are $\mathbf{x_1, x_2, \ldots, x_p}$. This can be written as $\mathbf{P(C_k | x_1, \ldots, x_p)}$.

Likelihood

Class Prior Probability

$$P(c\,|\,x)=\frac{P(x\,|\,c)P(c)}{P(x)}$$

Posterior Probability

Predictor Prior Probability

$$P(c\,|\,X) = P(x_1\,|\,c)\times P(x_2\,|\,c)\times\cdots\times P(x_n\,|\,c)\times P(c)$$

- *P*(*c*|*x*) is the posterior probability of *class* (*target*) given *predictor* (*attribute*).
- *P*(*c*) is the prior probability of *class*.
- *P*(*x*|*c*) is the likelihood which is the probability of *predictor* given *class*.
- *P*(*x*) is the prior probability of *predictor*.

Naïve Bayes Theorem

EMPLOYEE ATTRITION CLASSIFICATION PROBLEM

X1	X2	X3	X4	X5	X6	X6	X7	X8	Label
Emp_Id	**Skill**	**Years_of_Exp**	**Years_with_Company**	**Latest_Rating**	**Salary**	**Market_alignment**	**Manager_watch**	**Productivity**	**Attrition**
1	Java	5	3	3.5	40000	Y	N	G	Y
2	ML	6	4	3	45000	Y	Y	A	N
3	Java	4	2	4	41000	N	N	VG	N
4	Java	6	2	3.5	46000	N	N	VG	N
5	ML	9	4	3.5	51000	Y	Y	G	Y
6	ML	5	1	3	38000	Y	Y	A	Y
7	DS	2	2	3.5	31000	Y	N	A	N
8	Python	6	3	3.5	48000	N	N	G	N
9	React	7	2	4	49000	Y	N	G	N
10	ML	5	1	4	39000	N	N	G	Y
11	DS	4	3	4.5	38000	N	Y	G	?

FEATURES / X

CLASS / LABEL

Naïve Bayes Assumption

Naive Bayes classifier assumes that the ***value of a particular feature is independent of the value of any other feature***, given the class variable.

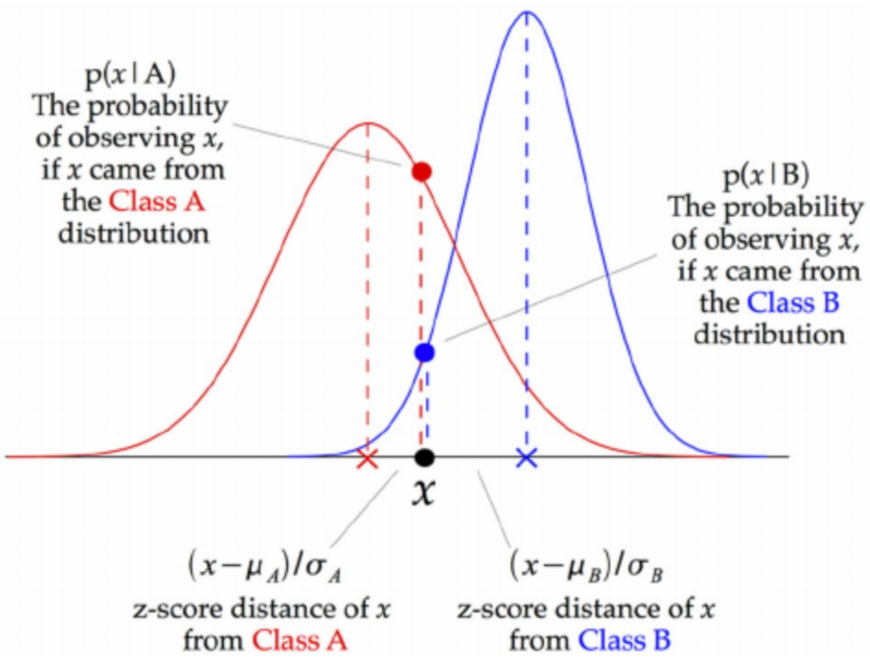
For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A Naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the colour, roundness, and diameter features. It’s a ***naïve assumption*** and hence the name.

Given that the Naïve Bayes classifier is very simple and computationally, the assumption of independence is not always realistic on our training data and hence not suitable in conditions where this assumption doesn’t hold valid.

Types of Naïve Bayes

Gaussian Naïve Bayes

In this classifier, the assumption is that data from each label is drawn from a simple ***Gaussian distribution***. You may recall that Gaussian distributions have no covariance between dimensions/features.



Example of Gaussian Data

- **Employee Salary**
- **Stock Price**
- **House/Car Price**

Types of Naïve Bayes

Multinomial Naïve Bayes

In Multinomial naive Bayes, the features are assumed to be generated from a simple **Multinomial Distribution**. ***The multinomial distribution describes the probability of observing counts among a number of categories, and thus multinomial naive Bayes is most appropriate for features that represent counts or count rates.***

The idea is precisely the same as Gaussian Naïve Bayes, except that instead of modelling the data distribution with the best-fit Gaussian, we model the data distribution with a best-fit multinomial distribution.

<u>Example Application</u>

One place where multinomial naive Bayes is often used is in **text classification**, where the features are **word counts** or word frequencies within the documents to be classified (say as a Spam or Ham or classified as Positive or Negative, etc). There are multiple NLP techniques of word features extraction like TF-IDF or Bag of Words).

Sparse word count features are used to classify these documents into categories or classes using Multinomial Naïve Bayes.

	the	red	dog	cat	eats	food
the red dog →	1	1	1	0	0	0
cat eats dog →	0	0	1	1	1	0
dog eats food →	0	0	1	0	1	1
red cat eats →	0	1	0	1	1	0

Points to Note

Because naive Bayesian classifiers make strict assumptions of independence about data, they generally do not perform very well always as data most always contain some degree of correlation. However,

- They are extremely fast for both training and prediction.
- Provide straightforward probabilistic prediction.
- Easily interpretable.
- Easy to use as there are very few or none tuneable parameters.

These advantages make Naive Bayesian classifier is a very good choice as initial baseline classification for classification problems in hand. Based on the data, they might throw very good results. Otherwise, you may want to scout for more sophisticated classifiers.

Generally, Naïve Bayes works well in following conditions:

- Naive assumptions are met in the Data
- Classes or categories are well-separated, a complex model is not needed
- For high-dimensional data

Naïve Bayes - Calculation

	Weather	Temparature	Humidity	Windy	Play Golf
1	Rainy	Warm	High	False	No
2	Rainy	Warm	High	True	No
3	Overcast	Warm	High	False	Yes
4	Sunny	Moderate	High	False	Yes
5	Sunny	Cool	Normal	False	Yes
6	Sunny	Cool	Normal	True	No
7	Overcast	Cool	Normal	True	Yes
8	Rainy	Moderate	High	False	No
9	Rainy	Cool	Normal	False	Yes
10	Sunny	Moderate	Normal	False	Yes
11	Rainy	Moderate	Normal	True	Yes
12	Overcast	Moderate	High	True	Yes
13	Overcast	Warm	Normal	False	Yes
14	Sunny	Moderate	High	True	No

The Problem: We have data various weather parameter data and a decision on whether or not it is it is conducive to play Golf. The features are Weather, Humidity, Windy and Temperature.

We are to create a Naïve Bayes model from this data to determine for a give condition, it is conducive to play Golf or not.

Weather	Temparature	Humidity	Windy	Play Golf
Rainy	Warm	High	False	?

Naïve Bayes - Calculation

Frequency Table

		Play Golf	
		Yes	No
Weather	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

		Play Golf	
		Yes	No
Temparature	Warm	2	2
	Moderate	4	2
	Cool	3	1

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

Likelyhood Table

		Play Golf	
		Yes	No
Weather	Sunny	3/9	2/5
	Overcast	4/9	0/5
	Rainy	2/9	3/5

		Play Golf	
		Yes	No
Temparature	Warm	2/9	2/5
	Moderate	4/9	2/5
	Cool	3/9	1/5

		Play Golf	
		Yes	No
Humidity	High	3/9	4/5
	Normal	6/9	1/5

		Play Golf	
		Yes	No
Windy	False	6/9	2/5
	True	3/9	3/5

		Play Golf	
		Yes	No
Weather	Sunny	0.33	0.40
	Overcast	0.44	0.00
	Rainy	0.22	0.60

		Play Golf	
		Yes	No
Temparature	Warm	0.22	0.40
	Moderate	0.44	0.40
	Cool	0.33	0.20

		Play Golf	
		Yes	No
Humidity	High	0.33	0.80
	Normal	0.67	0.20

		Play Golf	
		Yes	No
Windy	False	0.67	0.40
	True	0.33	0.60

Naïve Bayes - Calculation

Weather	Temparature	Humidity	Windy	Play Golf	Yes	No
Rainy	Warm	High	False	?	**0.24198**	**0.94080**
					20.46%	**79.54%**

P(Yes | X) = P(Rainy | Yes) x P(Warm | Yes) x P(High | Yes) x P(False | Yes) x P(Yes) / P(X)

P(No | X) = P(Rainy | No) x P(Warm | No) x P(High | No) x P(False | No) x P(No) / (Rainy Warm High False)

RECAP …

Likelihood

Class Prior Probability

$$P(c|x)=\frac{P(x|c)P(c)}{P(x)}$$

Posterior Probability

Predictor Prior Probability

- $P(c|x)$ is the posterior probability of *class* (*target*) given *predictor* (*attribute*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

Gradient Descent

Python Demo

- **Gaussian Naïve Bayes**
- **Multinomial Naïve Bayes**
- **Project**

Hope you have liked this Video.
Please help us by providing your Ratings and Comments for this Course!

Thank You!!
Manas Dasgupta

Happy Learning!!

Fully Editable Icon Sets: A

You can Resize without losing quality

You can Change Fill Color &
Line Color

FREE
PPT TEMPLATES

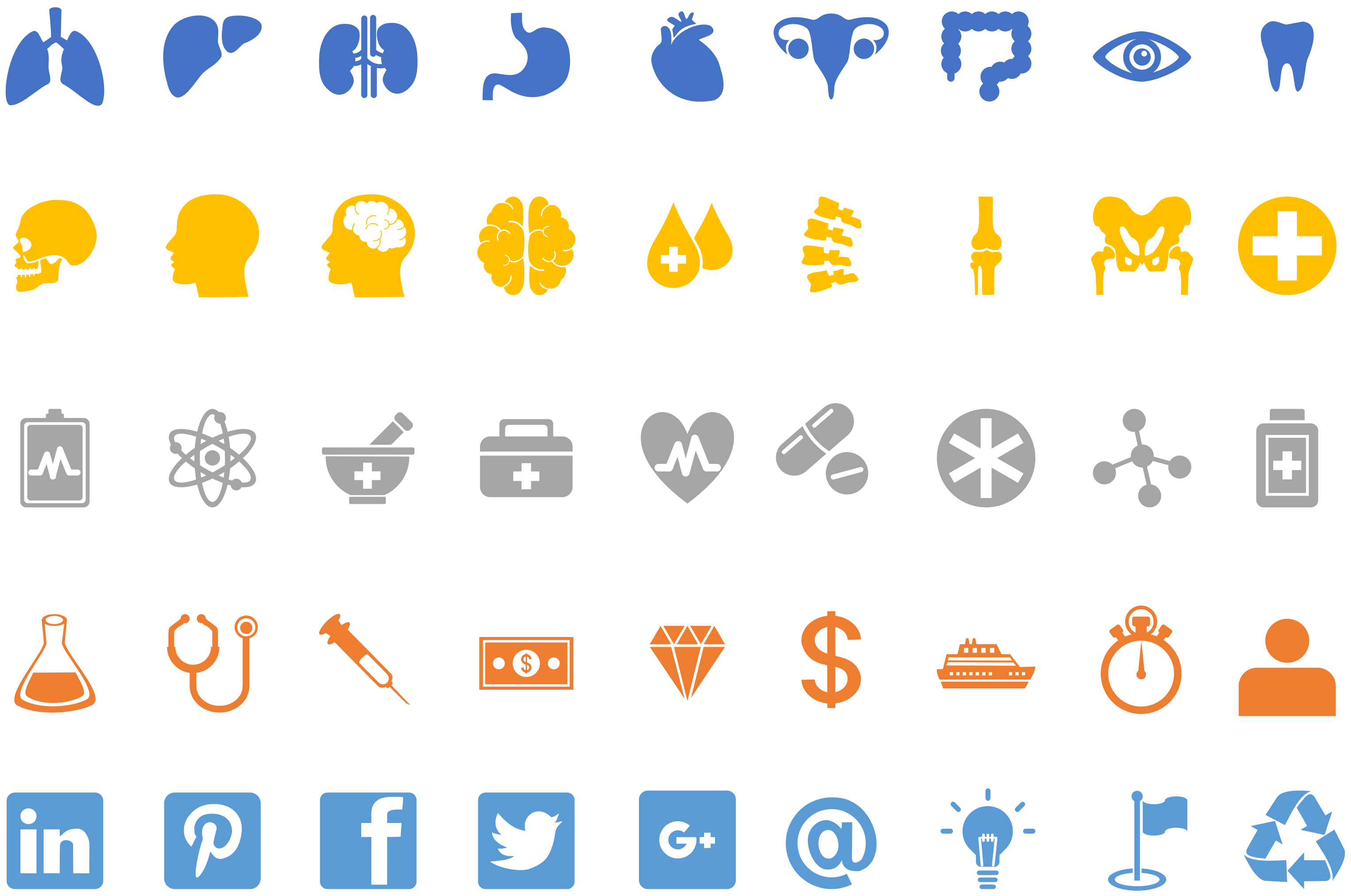
Fully Editable Icon Sets: B

You can Resize without losing quality

You can Change Fill Color &
Line Color

FREE
PPT TEMPLATES

www.allppt.com



Fully Editable Icon Sets: C

You can Resize without losing quality

You can Change Fill Color & Line Color

FREE

PPT TEMPLATES

www.allppt.com