Naïve Bayes

Mon, 23 May 2022 at 22:40

#bayes #theorem #classification

Overview

DEFINITION

> " ***naïve Bayes classifiers*** *are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features (see Bayes classifier)" wikipedia*

It is considered *naïve* as it does not take into consideration conditional dependence. in other terms, the pretense, order or any relationship between two features is not considered. It answers the basic question of what is the probability of class y, given set of features X

INPUT

Can be both discrete or continuous

OUTPUT

A probability score and a class label based on the highest probability score.

Use Cases

Used in text classification. Examples:

- Spam filtering
- Fraud detection
- Sentiment analysis

Bayesian Theorem Terminology

1. *Posteriori Probability* → The probability of an event occurring after taking into consideration new information (the prediction)

$$P(H|X) = \frac{P(X|H) \times P(H)}{P(X)}$$

2. *Prior Probability* → The initial probability of observed data
3. *Evidence* → Probability that sample data is observed
4. *Likelihood* → Conditional probability, the probability of observing the sample X given that the hypothesis holds

ALGORITHM

1. *Purity* → probability of the corresponding class based on the node's decision
   1. 100% **purity** when all of the node's data belongs to `one` class
   2. 100% **impurity** when the node's data is evenly split *same records of each class*
2. *Splitting rules* → defines how a decision tree is split

Classification by Maximum Posteriori

First: calculate the prior probability

Given a training set and their associated class labels, calculate the prior probability $P(c_i)$ for each class

Second: calculate the posteriori probability

For each attribute x_j calculate its posterior probability for every class.

The goal is to maximize $P(c_i|X)$ The formula is

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

Under the assumption that the attributes are mutually exclusive

$$P(X|C_i) = \prod_{k=1} P(x_k|Ci)$$

Meaning for attribute X that is categorical with three types X=1, X=2, X=3 we can find $P(X=1|C_i)$ by finding all the tuples in C_i with the value X=1 divided by total number of tuples in C_i

Third: assign the class to the highest probability

Once you calculate $P(H|X)$ for every class. choose the highest probability as the final classification

Pros & Cons

Reasons to Choose	**Cautions**
Handles missing values	Sensitive to correlated variables
Robust to irrelevant variables	Numeric variables have to be discrete
Easy to implement and score	Not good at estimating probabilities
Resistant to over fitting	
Computationally efficient	