**Naive Bayes Classifiers**

A Naive Bayes classifiers, a family of algorithms based on Bayes’ Theorem. Despite the “naive” assumption of feature independence, these classifiers are widely utilized for their simplicity and efficiency in machine learning.

**What is Naive Bayes Classifiers?**

Naive Bayes classifiers are a collection of classification algorithms based on Bayes’ Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other. To start with, let us consider a dataset.

One of the most simple and effective classification algorithms, the Naïve Bayes classifier aids in the rapid development of machine learning models with rapid prediction capabilities.

Naïve Bayes algorithm is used for classification problems. It is highly used in text classification. In text classification tasks, data contains high dimension (as each word represent one feature in the data). It is used in spam filtering, sentiment detection, rating classification etc. The advantage of using naïve Bayes is its speed. It is fast and making prediction is easy with high dimension of data.

This model predicts the probability of an instance belongs to a class with a given set of feature value. It is a probabilistic classifier. It is because it assumes that one feature in the model is independent of existence of another feature. In other words, each feature contributes to the predictions with no relation between each other. In real world, this condition satisfies rarely. It uses Bayes theorem in the algorithm for training and prediction.

**Why it is Called Naive Bayes?**

The “Naive” part of the name indicates the simplifying assumption made by the Naïve Bayes classifier. The classifier assumes that the features used to describe an observation are conditionally independent, given the class label. The “Bayes” part of the name refers to Reverend Thomas Bayes, an 18th-century statistician and theologian who formulated Bayes’ theorem.

Consider a fictional dataset that describes the weather conditions for playing a game of golf. Given the weather conditions, each tuple classifies the conditions as fit(“Yes”) or unfit(“No”) for playing golf. Here is a tabular representation of our dataset.

| **Outlook** | **Temperature** | **Humidity** | **Windy** | **Play Golf** |
| --- | --- | --- | --- | --- |
| 0 | Rainy | Hot | High | False | No |
| 1 | Rainy | Hot | High | True | No |
| 2 | Overcast | Hot | High | False | Yes |
| 3 | Sunny | Mild | High | False | Yes |
| 4 | Sunny | Cool | Normal | False | Yes |
| 5 | Sunny | Cool | Normal | True | No |
| 6 | Overcast | Cool | Normal | True | Yes |
| 7 | Rainy | Mild | High | False | No |
| 8 | Rainy | Cool | Normal | False | Yes |
| 9 | Sunny | Mild | Normal | False | Yes |
| 10 | Rainy | Mild | Normal | True | Yes |
| 11 | Overcast | Mild | High | True | Yes |
| 12 | Overcast | Hot | Normal | False | Yes |
| 13 | Sunny | Mild | High | True | No |

The dataset is divided into two parts, namely, feature matrix and the response vector.

Feature matrix contains all the vectors(rows) of dataset in which each vector consists of the value of dependent features. In above dataset, features are ‘Outlook’, ‘Temperature’, ‘Humidity’ and ‘Windy’.

Response vector contains the value of class variable(prediction or output) for each row of feature matrix. In above dataset, the class variable name is ‘Play golf’.

**Assumption of Naive Bayes**

The fundamental Naive Bayes assumption is that each feature makes an:

Feature independence: The features of the data are conditionally independent of each other, given the class label.

Continuous features are normally distributed: If a feature is continuous, then it is assumed to be normally distributed within each class.

Discrete features have multinomial distributions: If a feature is discrete, then it is assumed to have a multinomial distribution within each class.

Features are equally important: All features are assumed to contribute equally to the prediction of the class label.

No missing data: The data should not contain any missing values.

With relation to our dataset, this concept can be understood as:

* We assume that no pair of features are dependent. For example, the temperature being ‘Hot’ has nothing to do with the humidity or the outlook being ‘Rainy’ has no effect on the winds. Hence, the features are assumed to be independent.
* Secondly, each feature is given the same weight(or importance). For example, knowing only temperature and humidity alone can’t predict the outcome accurately. None of the attributes is irrelevant and assumed to be contributing equally to the outcome.

Advantages of Naive Bayes Classifier

* Easy to implement and computationally efficient.
* Effective in cases with a large number of features.
* Performs well even with limited training data.
* It performs well in the presence of categorical features.
* For numerical features data is assumed to come from normal distributions

Disadvantages of Naive Bayes Classifier

* Assumes that features are independent, which may not always hold in real-world data.
* Can be influenced by irrelevant attributes.
* May assign zero probability to unseen events, leading to poor generalization.

Applications of Naive Bayes Classifier

* Spam Email Filtering: Classifies emails as spam or non-spam based on features.
* Text Classification: Used in sentiment analysis, document categorization, and topic classification.
* Medical Diagnosis: Helps in predicting the likelihood of a disease based on symptoms.
* Credit Scoring: Evaluates creditworthiness of individuals for loan approval.
* Weather Prediction: Classifies weather conditions based on various factors.