09 Naive Bayes

- Naive Bayes is a simple yet powerful probabilistic classifier based on Bayes' Theorem
- It is particularly useful for tasks like text classification, spam detection, sentiment analysis, and more

Key Concepts of Naive Bayes

1. **Bayes' Theorem**

- Bayes' Theorem describes the probability of an event based on prior knowledge of conditions related to the event

  $P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$
- (P(A|B)) is the posterior probability: the probability of event (A) occurring given that (B) is true
- (P(B|A)) is the likelihood: the probability of event (B) occurring given that (A) is true
- (P(A)) is the prior probability of event (A)
- (P(B)) is the prior probability of event (B)

2. **Naive Assumption**
   - The "naive" part of Naive Bayes comes from the assumption that all features are independent of each other given the class label
   - This simplifies the computation since you can calculate the probability of each feature independently
3. **Classification**
   - For a given set of features X and a set of possible classes C Naive Bayes assigns the class C to X that maximizes the posterior probability

     $C = \arg\max_{C_k} P(C_k) \prod_{i=1}^{n} P(x_i|C_k)$

Types of Naive Bayes Classifiers

1. **Gaussian Naive Bayes**
   - Used when the features are continuous and are assumed to follow a normal (Gaussian) distribution
   - It calculates the probability using the probability density function of a Gaussian distribution
2. **Multinomial Naive Bayes**

- Used for discrete data, especially in text classification, where the features are word counts or term frequencies
- It models the probability of features (words) based on a multinomial distribution

3. **Bernoulli Naive Bayes**
    - Used for binary/boolean features, where each feature is either present or absent
    - It is particularly useful for binary text classification (e.g., spam vs. non-spam)

Advantages

- Simple to implement
- Works well with high-dimensional data (e.g., text)
- Fast and efficient for both training and prediction

Disadvantages

- The strong independence assumption rarely holds true in real-world data, which can reduce accuracy
- It assumes all features contribute equally to the prediction, which might not be the case

01 Multinomial Naive Bayes

- Multinomial Naive Bayes assumes that the data (features) follow a multinomial distribution, which is a generalization of the binomial distribution
- In the context of text classification, the classifier computes the probability of a document belonging to a particular class (e.g., spam or not spam) by considering the frequency of words (features) within the document

For a given document ($d = (x_1, x_2, \dots, x_n)$), where (x_i) represents the frequency of word (i), the probability that the document belongs to class (C_k) is given by:

$P(C_k|d) \propto P(C_k) \prod_{i=1}^{n} P(x_i|C_k)^{x_i}$

- ($P(C_k)$) is the prior probability of class (C_k)
- ($P(x_i | C_k)$) is the probability of word (x_i) occurring in class (C_k)
- (x_i) is the count of word (i) in the document

02 Laplace Smoothing

- **Laplace Smoothing** is a technique used to handle the issue of zero probabilities in Naive Bayes classifiers

- This problem arises when a word in the test data hasn't appeared in the training data for a particular class, leading to a zero probability for the entire expression
- Laplace smoothing adds a small value (typically 1) to each word count, ensuring that no probability is ever zero
- If (n) is the total number of words in the vocabulary, and (\alpha) is the smoothing parameter (usually (\alpha = 1)), the smoothed probability of a word (x_i) in class (C_k) is calculated as:

  $$P(x_i|C_k) = \frac{count(x_i, C_k) + \alpha}{\sum_{j=1}^{n}(count(x_j, C_k) + \alpha)}$$

03 Bernoulli Naive Bayes

- In Bernoulli Naive Bayes, each feature is a binary value indicating whether a particular word or feature is present in the document
- The classifier calculates the probability of a document belonging to a class based on whether the words/features appear in the document

For a document (d = (x_1, x_2, \dots, x_n)) where (x_i) is binary:

$$P(C_k|d) \propto P(C_k) \prod_{i=1}^{n} P(x_i|C_k)^{x_i} \times (1 - P(x_i|C_k))^{(1-x_i)}$$

- (P(x_i | C_k)) is the probability that word (x_i) appears in documents of class (C_k).
- ((1 - P(x_i | C_k))) accounts for the absence of the word (x_i).

04 Gaussian Naive Bayes

- **Gaussian Naive Bayes** is used when the features are continuous rather than discrete
- It assumes that the features follow a normal (Gaussian) distribution
- The classifier assumes that the continuous features associated with each class are distributed according to a Gaussian distribution
- For a feature (x_i), given a class (C_k), the probability (P(x_i | C_k)) is calculated using the Gaussian probability density function:

  $$P(x_i|C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}\right)$$