**Assignment No. 4**

**Problem Statement:** Understand and implement the Naïve Bayes classification algorithm.

**Objective:**

1. Understand the Theory – Learn the mathematical background of the Naïve Bayes algorithm, including Bayes' Theorem and conditional probabilities.
2. Implement Naïve Bayes – Apply the algorithm to a dataset and analyze its performance.
3. Evaluate Performance – Measure the accuracy, precision, recall, and F1-score of the classifier.

**Prerequisite :**

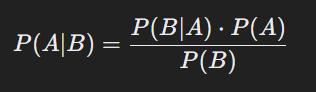
1. A Python environment with essential libraries like pandas, numpy, matplotlib, seaborn, and scikit-learn.
2. Basic knowledge of Python, statistics, and machine learning principles.
3. Statistics Knowledge – Understanding of probability, conditional probability, and Bayes' Theorem.
4. Machine Learning Principles – Familiarity with classification techniques and model evaluation metrics.

**Theory :**

Naïve Bayes is a classification algorithm based on **Bayes’ Theorem**, which calculates the probability of a class given certain features. It is called "naïve" because it assumes that all features are **independent** of each other, which may not always be true in real-world datasets. Despite this simplification, it performs well in many applications.

### ****Bayes’ Theorem****

The foundation of Naïve Bayes is **Bayes' Theorem**, which states:



Where:

* P(A∣B) = Probability of event A occurring given that event B has occurred (posterior probability).
* P(B∣A) = Probability of event B occurring given that event A has occurred (likelihood).
* P(A) = Prior probability of event A occurring.
* P(B) = Total probability of event B occurring.

1. **Assumptions of Naïve Bayes**
2. **Feature Independence** – Each feature contributes **independently** to the probability of a class label.
3. **Equal Importance of Features** – All features are given equal importance in predicting the output.
4. **Conditional Independence** – Given the class label, the features do not depend on each other.

**3. Types of Naïve Bayes Classifiers**

There are three main types of Naïve Bayes classifiers:

1. **Gaussian Naïve Bayes (GNB)** – Assumes that features follow a normal (Gaussian) distribution. Used for continuous numerical data.
2. **Multinomial Naïve Bayes (MNB)** – Suitable for classification with **discrete count data**, commonly used in text classification.
3. **Bernoulli Naïve Bayes (BNB)** – Works with **binary/boolean features**, mainly used in spam detection or sentiment analysis.

**4. Steps in Naïve Bayes Classification**

1. **Data Preprocessing** – Load the dataset, clean missing values, and prepare feature-target variables.
2. **Calculate Prior Probabilities** – Compute P(A) for each class in the dataset.
3. **Compute Likelihood** – Calculate P(B∣A) based on the type of Naïve Bayes model used (Gaussian, Multinomial, or Bernoulli).
4. **Apply Bayes' Theorem** – Compute the **posterior probability** P(A∣B) for each class and assign the highest probability class to the data point.
5. **Evaluate Performance** – Use accuracy, precision, recall, and F1-score to assess the classifier's performance.

**5. Advantages of Naïve Bayes**

1. **Fast and Efficient** – Works well with large datasets and high-dimensional data.
2. **Handles Missing Data** – Can work well even if some features have missing values.
3. **Performs Well with Small Data** – Requires less training data compared to other classifiers.
4. **Interpretable and Simple** – Easy to implement and understand.
5. **Works Well in Text Classification** – Commonly used in spam detection, sentiment analysis, and document classification.

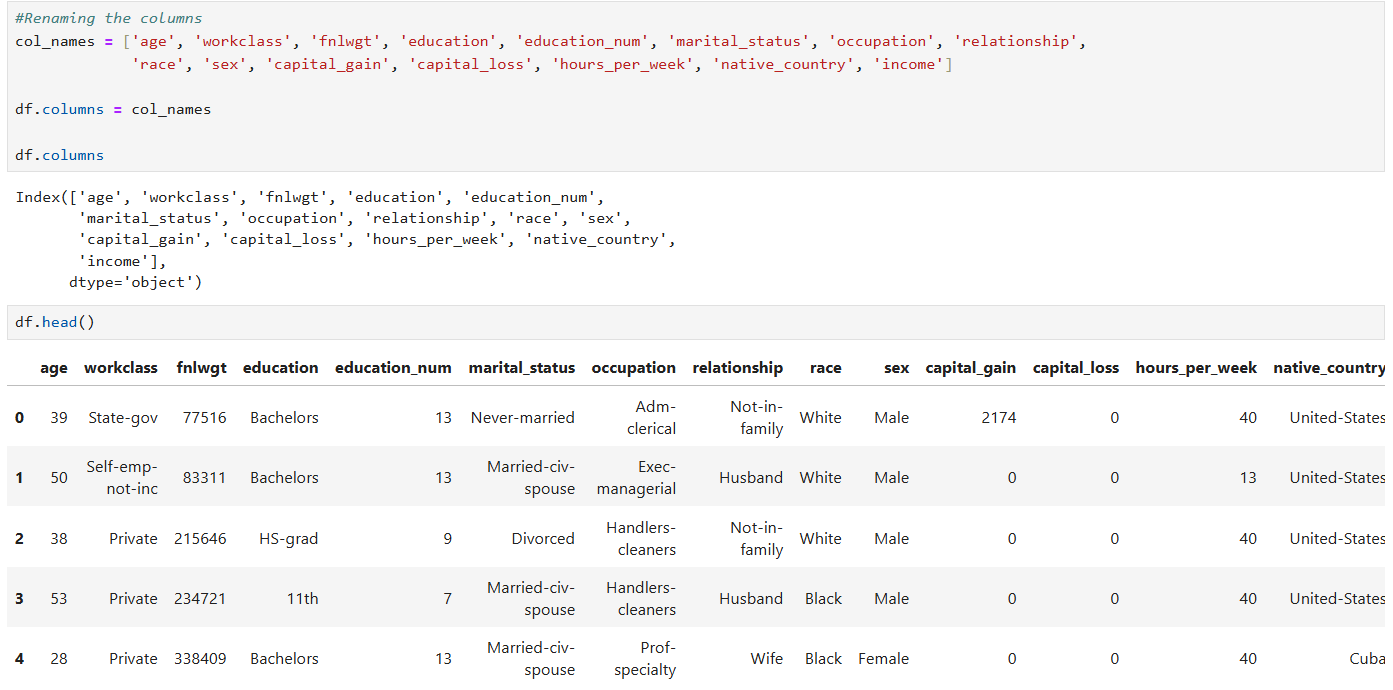
**6. Disadvantages of Naïve Bayes**

1. **Strong Feature Independence Assumption** – Often unrealistic in real-world data.
2. **Zero Probability Issue** – If a category is missing in the training dataset, it assigns zero probability (solved using **Laplace smoothing**).
3. **Poor Performance on Highly Correlated Features** – If features are dependent, it may give incorrect classifications.
4. **Limited for Complex Datasets** – Not ideal for datasets where relationships between features are important.

**7. Applications of Naïve Bayes**

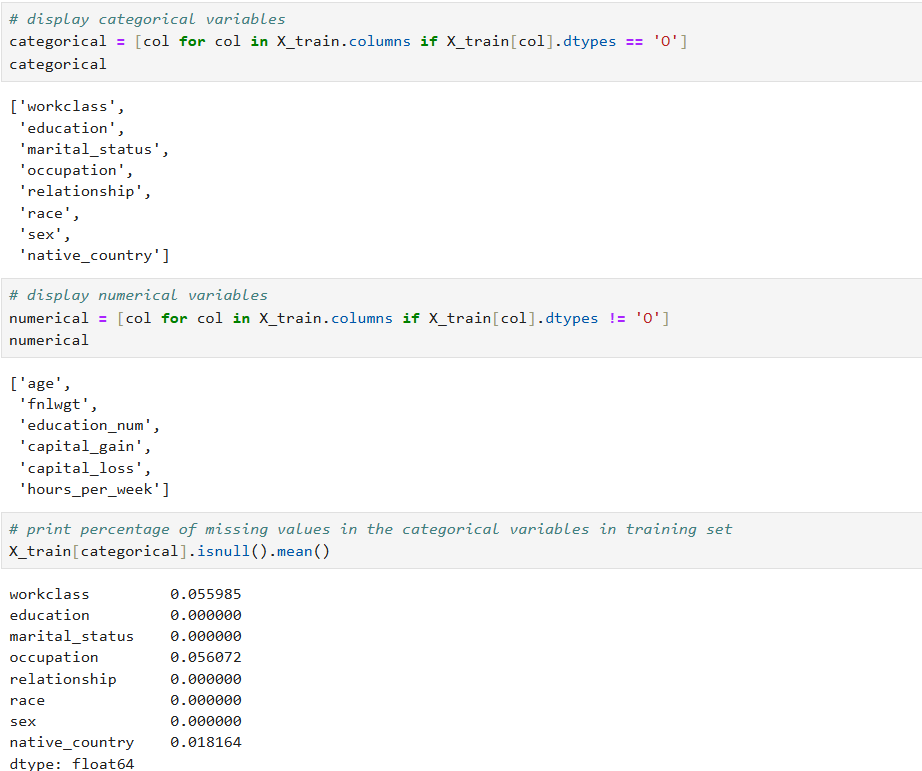
1. **Spam Detection** – Classifies emails as spam or not spam based on word frequency.
2. **Sentiment Analysis** – Determines the sentiment of text (positive, negative, neutral).
3. **Medical Diagnosis** – Predicts diseases based on patient symptoms.
4. **Credit Scoring** – Assesses credit risk in finance.
5. **Recommendation Systems** – Suggests products or content based on user behavior.
6. **Code & Output**

****

****

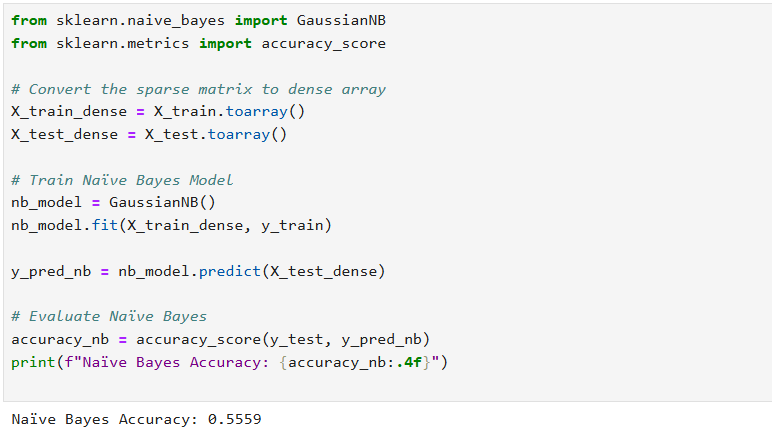
****

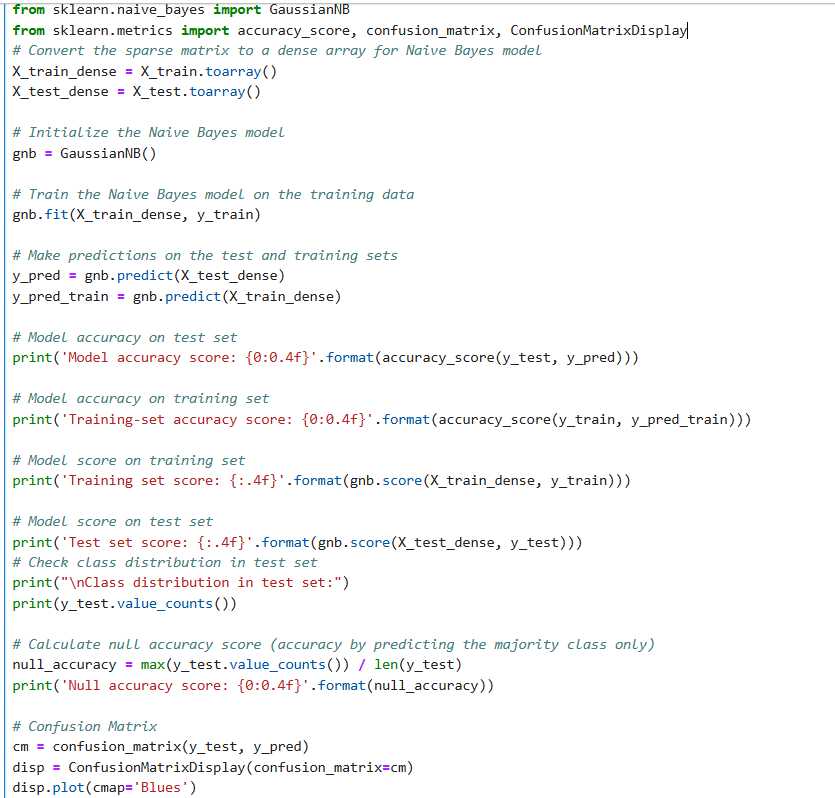
****

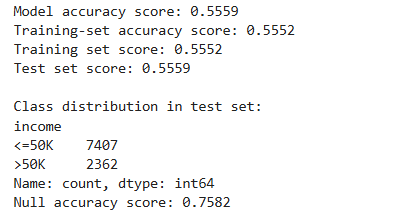
****

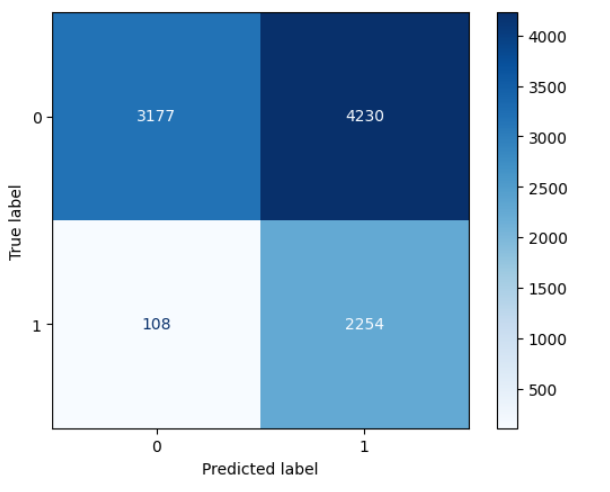
****

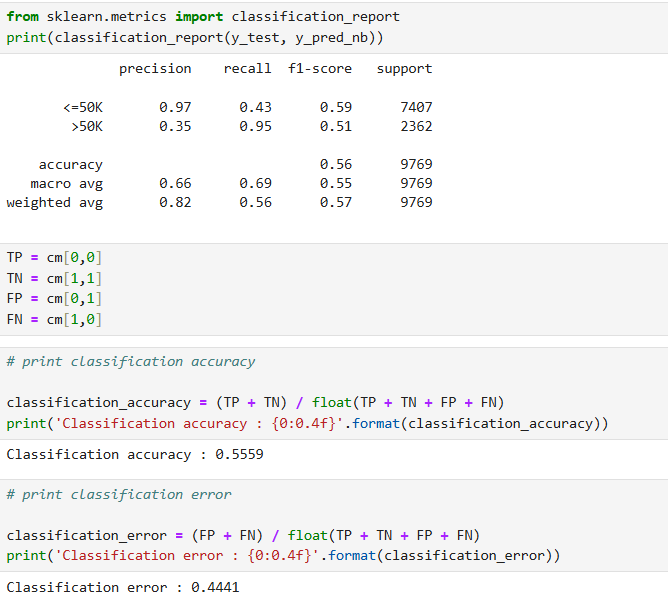
****

****

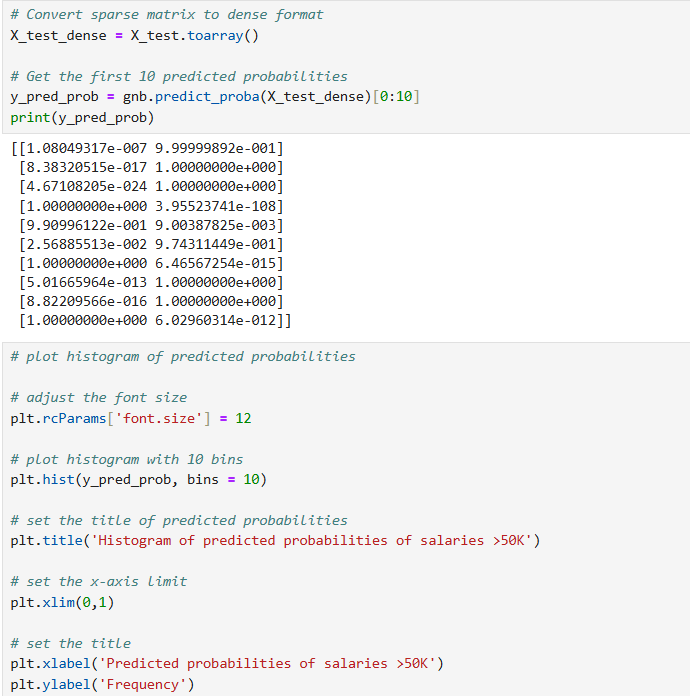


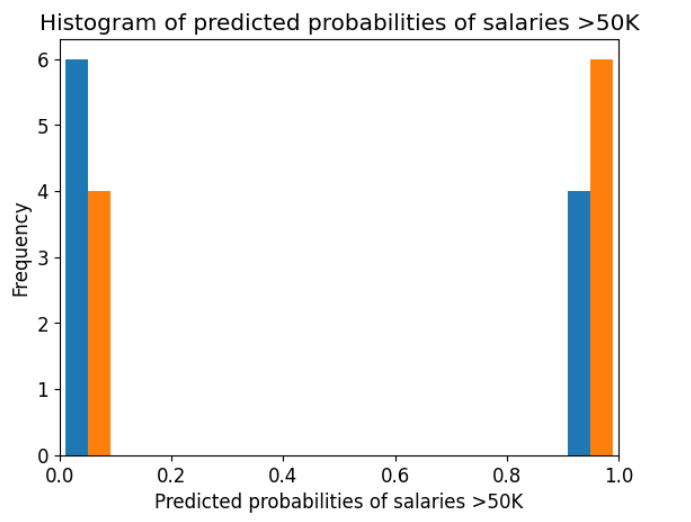












## Github: <https://github.com/dnyaneshwardhere/ML>

## **Conclusion:**

Gaussian Naïve Bayes is an effective and simple classification algorithm for numerical datasets. Despite its **assumption of feature independence**, it performs well in many real-world applications, especially when the data is normally distributed. While it has limitations, such as sensitivity to non-Gaussian distributions and the independence assumption, its efficiency, simplicity, and effectiveness make it a valuable tool in machine learning.