**Aim:** Apply the Naive Bayes machine learning algorithm for classification tasks and assess accuracy, precision, and recall.

**Software Used:** Google Collaborative Python

**Theory:**

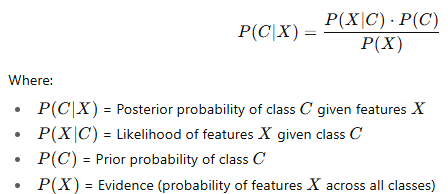
**Naive Bayes Classifier:**

Naive Bayes is a classification algorithm that uses probability to predict which category a data point belongs to, assuming that all features are unrelated.

**Naive Bayes** is a **supervised machine learning algorithm** based on **Bayes’ Theorem** and is primarily used for **classification tasks**. It is called "naive" because it assumes that all features in the dataset are **independent** of each other given the class label, which in real-world scenarios is rarely true, but the algorithm still performs well in many applications.

The algorithm predicts the probability that a given data point belongs to a particular class and assigns it to the class with the highest posterior probability.

**Bayes’ Theorem:**



**Naive Assumption:**

* All features contribute independently to the probability of the class.
* This simplifies computation and makes the algorithm fast even for large datasets.

**Types of Naive Bayes Model**

**1. Gaussian Naive Bayes:** In [Gaussian Naive Bayes](https://www.geeksforgeeks.org/machine-learning/gaussian-naive-bayes/), continuous values associated with each feature are assumed to be distributed according to a Gaussian distribution. A Gaussian distribution is also called [Normal distribution](https://en.wikipedia.org/wiki/Normal_distribution) When plotted, it gives a bell shaped curve which is symmetric about the mean of the feature values as shown below:

**2. Multinomial Naive Bayes;** [Multinomial Naive Bayes](https://www.geeksforgeeks.org/machine-learning/multinomial-naive-bayes/)is used when features represent the frequency of terms (such as word counts) in a document. It is commonly applied in text classification, where term frequencies are important.

**3. Bernoulli Naive Bayes:** [Bernoulli Naive Bayes](https://www.geeksforgeeks.org/machine-learning/bernoulli-naive-bayes/) deals with binary features, where each feature indicates whether a word appears or not in a document. It is suited for scenarios where the presence or absence of terms is more relevant than their frequency. Both models are widely used in document classification tasks

**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**

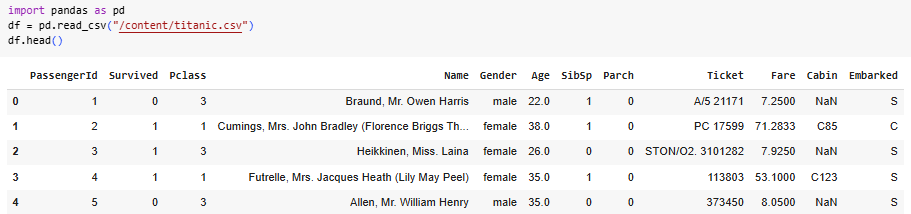
* Assumes that features are independent, which may not always hold in real-world data.
* Can be influenced by irrelevant attributes.
* May assign zero probability to unseen events, leading to poor generalization.

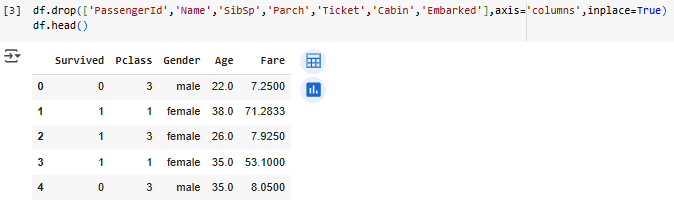
**Applications of Naive Bayes Classifier**

* 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.

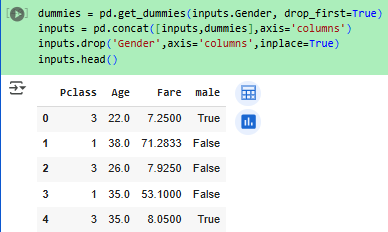
**Output:**

**1)titanic.csv**

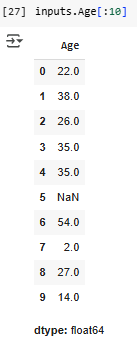
****

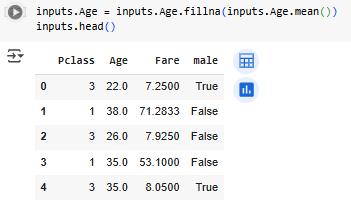
****

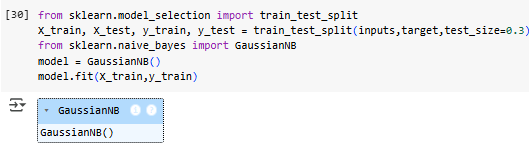
****

****

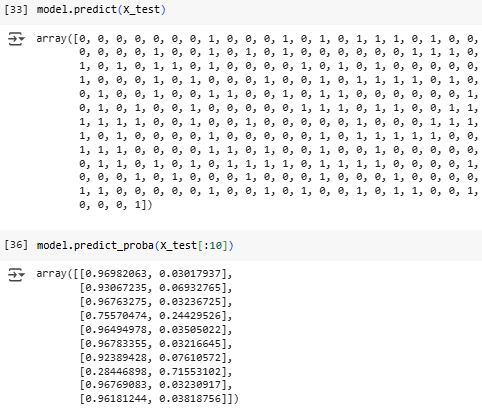
****

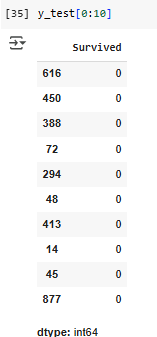
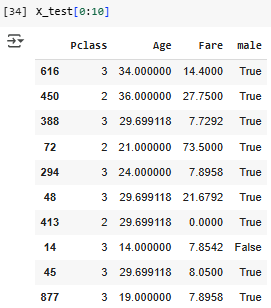
****

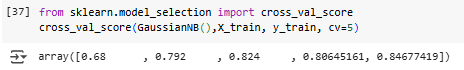
****

****

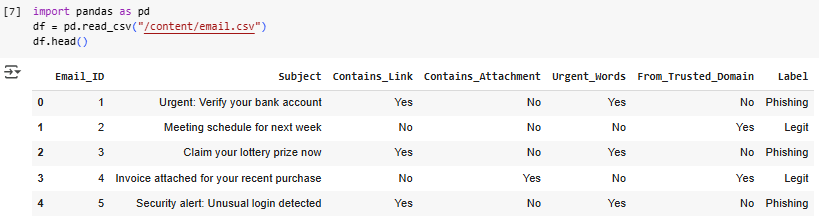
****

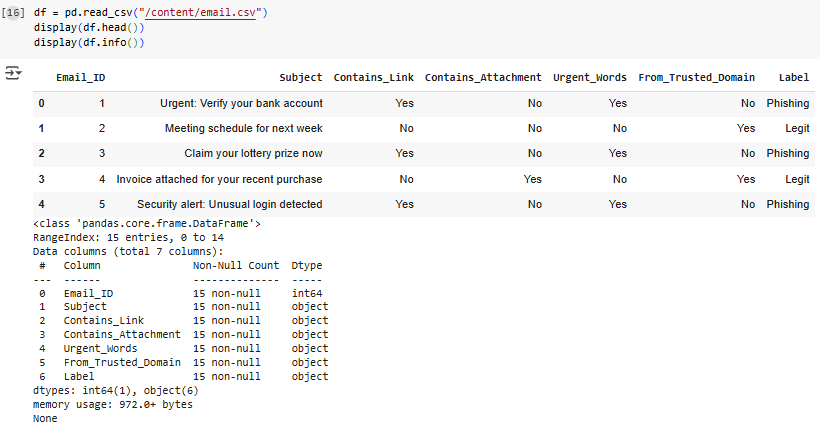
****

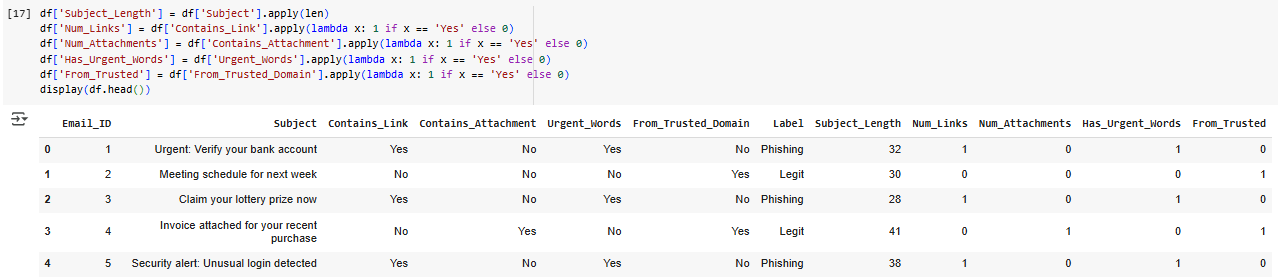
****

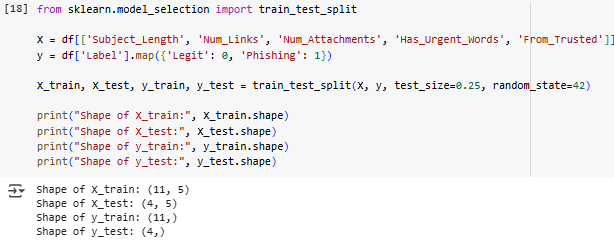
****

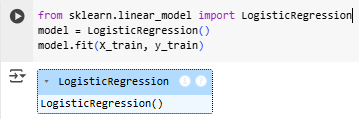
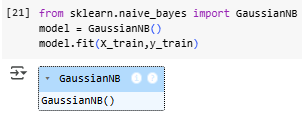
**2)email.csv**

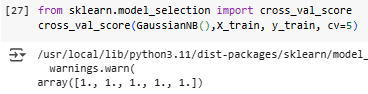
****

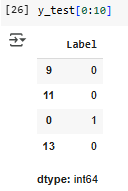
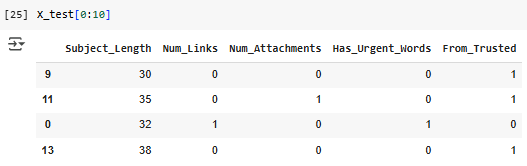
****

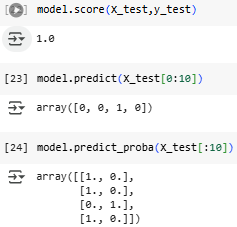
****

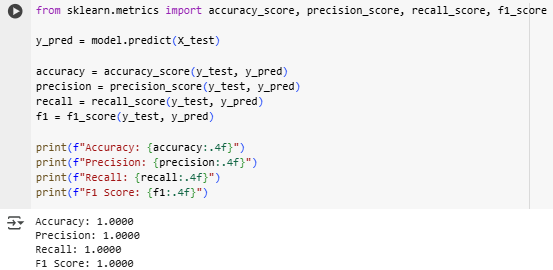
****

****

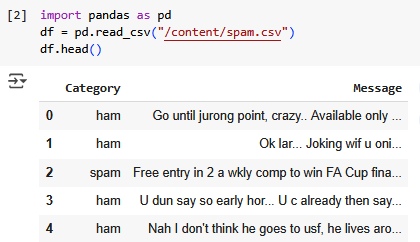
****

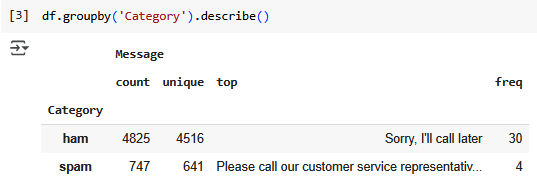
****

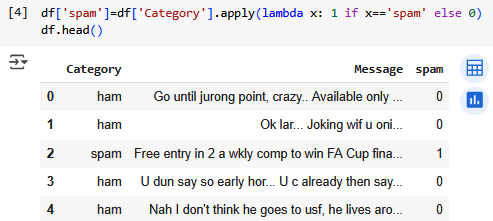
****

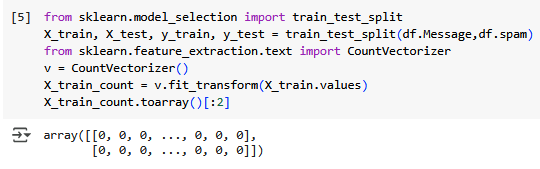
****

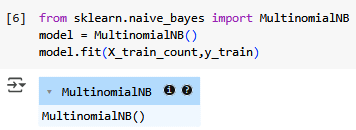
**3)spam.csv**

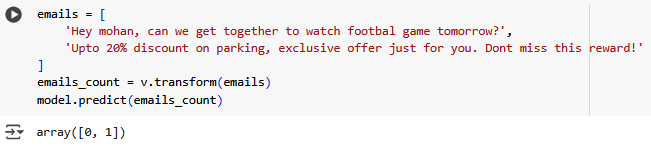
****

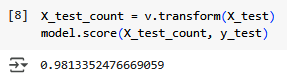
****

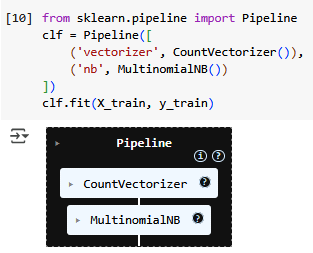
****

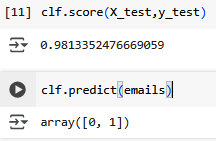
****

****

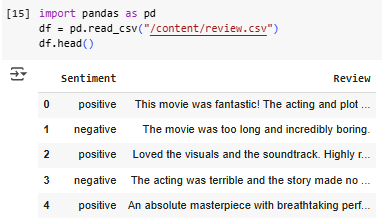
****

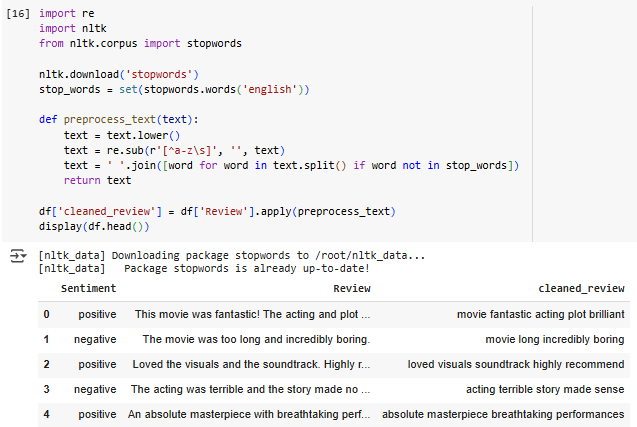
****

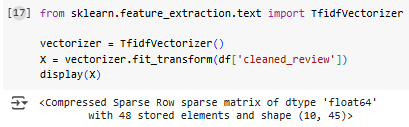
****

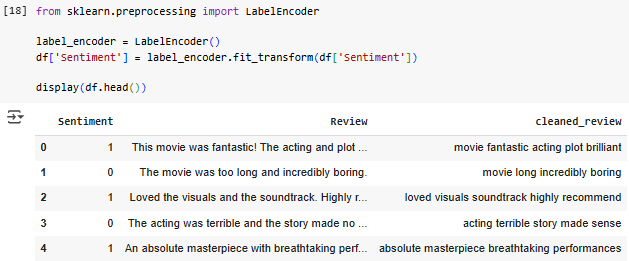
****

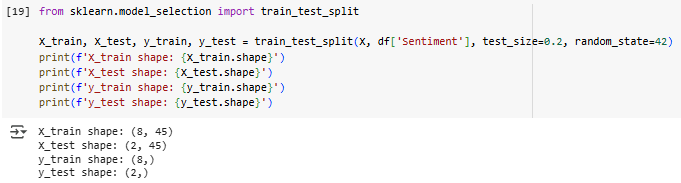
**4)review.csv**

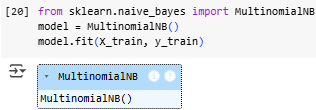
****

****

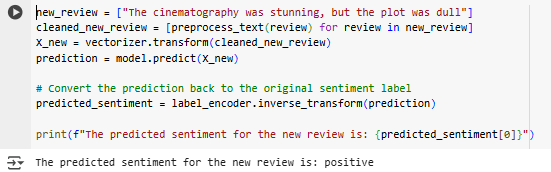
****

****

****

****

****

****

**Conclusion:** Hence, through this experiment, we learned how to apply the Naive Bayes algorithm for classification tasks, train it using different variations like Gaussian and Multinomial Naive Bayes, and evaluate the model using metrics such as accuracy, precision, and recall. This process is essential for building efficient, probabilistic classification models that work well with large and high-dimensional datasets.