Experiment No. 6

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

Software tools: Google Colab, Python Libraries(Pandas,Scikit-learn)

Theory:

This experiment focuses on the application of the Naive Bayes machine learning algorithm using two distinct datasets: the Titanic dataset and an email spam classification dataset. The objective is to demonstrate how Naive Bayes can be effectively utilized for classification tasks, showcasing its strengths and limitations in real-world scenarios.

Naive Bayes (NB) is a simple, probabilistic machine learning algorithm primarily used for classification tasks. It's based on Bayes' theorem and assumes independence between features.Naive Bayes is a family of supervised machine learning algorithms based on Bayes' Theorem, primarily used for classification tasks. It is particularly effective in scenarios where the dimensionality of the input data is high, such as text classification problems. Here’s a detailed overview:

Bayes' Theorem:
At its core, Naive Bayes relies on Bayes' Theorem, which describes the probability of a class given certain features. The theorem can be expressed mathematically as:

$$P(Y|X) = \frac{P(X|Y) \cdot P(Y)}{P(X)}$$

Where:

- $P(Y|X)$ is the **posterior probability**: the probability of class Y given the features X.
- $P(X|Y)$ is the **likelihood**: the probability of features X given class Y.
- $P(Y)$ is the **prior probability**: the initial probability of class Y.
- $P(X)$ is the **evidence**: the total probability of features X.

Evaluation Metrics :

- **Accuracy** measures the overall correctness of the model.

  Accuracy = Correct predictions / Total prediction

- **Precision** focuses on how many predicted positives are actually positive.
  Precision = TP / (TP + FP)
- **Recall** focuses on how many actual positives were correctly identified.
  Recall = TP / (TP + FN)

TF-IDF Vectorization

TF-IDF stands for **Term Frequency – Inverse Document Frequency**. It's a way to convert text into **numerical features** so machine learning models can understand it.

Advantages :

- Simplicity and Speed: Easy to implement and computationally efficient, making it suitable for large datasets.
- Effective with High Dimensional Data: Performs well in text classification tasks where the number of features (words) can be very large.
- Works Well with Limited Data: Requires less training data to estimate parameters compared to many other algorithms.

Limitations :

- Independence Assumption: The assumption that all features are independent can lead to inaccurate predictions when features are correlated.
- Zero Probability Problem: If a category has no training instances for a feature, it can lead to zero probabilities in predictions. This can be mitigated using techniques like Laplace smoothing.

Applications :

- Spam Detection: Classifying emails as spam or not spam based on their content.
- Sentiment Analysis: Determining sentiment polarity (positive, negative, neutral) from text data.
- Medical Diagnosis: Predicting diseases based on symptoms.
- Document Classification: Categorizing documents into predefined classes.

Conclusion :

We have implemented the Naive Bayes algorithm for classification tasks, and the models were able to achieve good performance in terms of accuracy, precision, and recall.