Sardar Patel Institute of Technology, Mumbai

Department of Electronics and Telecommunication Engineering

B.E. Sem-VII (2022-2023)

EC344 - Machine Learning and AI

**Experiment: Implement the Naïve-Bayes classifier**

**Name:**  **Roll No.**  **Date:**

**Objective**: Implement the naïve Bayesian Classifier model to classify a set of documents that you have assumed. Calculate the accuracy, precision, and recall for your data set.

**Outcomes**:

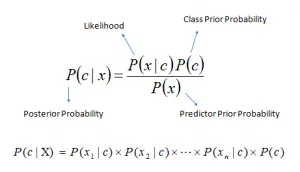
1. Find the conditional probabilities of attributes of the train data using Bayes theorem and follow the steps of the algorithm.
2. Apply the Naïve-Bayes algorithm to classify the given documents.
3. Apply Parameter smoothing for non-occurring values of attributes while calculation.
4. Find accuracy, precision, recall of the model for test data set.

**System Requirements:** Linux OS with Python and libraries or R or windows with MATLAB

**Theory :**

Naive Bayes algorithm is a classification technique based on Bayes’ Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:



* P(c|x) is the posterior probability of class (c, target) given predictor (x, attributes).
* P(c) is the prior probability of class.
* P(x|c) is the likelihood which is the probability of predictor given class.
* P(x) is the prior probability of predictor.

**Naive Bayes algorithm:**

Step 1: Convert the data set into a frequency table

Step 2: Create a likelihood table by finding the probabilities

Step 3: Calculate the posterior probability of each feature with respect to the class.

Step 4: If for a certain feature the probability evaluates to zero use feature smoothening for correction.



Step 5: Classify the example into the class for which the probability is highest.

**Performance parameters of the model :**

**Accuracy:** It is the ratio of number of correct predictions to the total number of input samples.

**Precision:** Precision is defined as the fraction of the examples which are actually positive among all the examples which we predicted positive.

**Recall:** We define recall as, among all the examples that actually positive, what fraction did we detect as positive?

**F1-score:** F1 Score is the Harmonic Mean between precision and recall.

**Confusion Matrix:** Confusion Matrix as the name suggests gives us a matrix as output and describes the complete performance of the model.

There are 4 important terms :

* True Positives: The cases in which we predicted YES and the actual output was also YES.
* True Negatives: The cases in which we predicted NO and the actual output was NO.
* False Positives: The cases in which we predicted YES and the actual output was NO.
* False Negatives: The cases in which we predicted NO and the actual output was YES.

**Dataset:**

**Code:**

**Output:**

To improve the performance of a Naive Bayes Classifier

* Use probabilities for feature selection
* Remove zero observations problem by using feature smoothening
* Remove redundant features

**Conclusion:**

* We learned how the Naive Bayes classifier uses posterior probability and feature smoothing to classify an example into a class.
* Using Sci-Kit we feature engineered the dataset creating a feature vector and count vector to determine the frequency of each word in the documents.
* We build the Naive Bayes model using the Multinomial classifier and generated the performance report of the classifier for calculating accuracy, precision, recall and creating the confusion matrix.