Experiment No. 6

Aim: Data Analytics III

Problem Statement:

1. Implement Simple Naïve Bayes classification algorithm using Python/R on iris.csv dataset.
2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.

Theory: Naive Bayes is a machine learning algorithm that is used by data scientists for classification. The naive Bayes algorithm works based on the Bayes theorem. Before explaining Naive Bayes, first, we should discuss Bayes Theorem. Bayes theorem is used to find the probability of a hypothesis with given evidence. This beginner-level article intends to introduce you to the Naive Bayes algorithm and explain its underlying concept and implementation.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

In this equation, using Bayes theorem, we can find the probability of A, given that B occurred. A is the hypothesis, and B is the evidence.

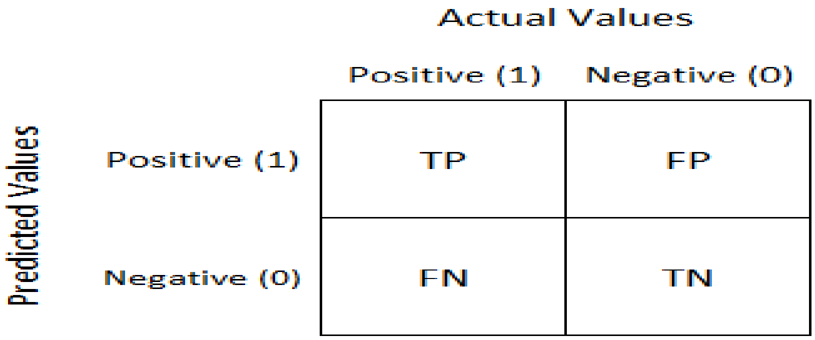
P(B|A) is the probability of B given that A is True.

P(A) and P(B) are the independent probabilities of A and B.

The Naive Bayes algorithm is a classification algorithm that is based on Bayes' theorem, which is a way of calculating the probability of an event based on its prior knowledge. The algorithm is called "naive" because it makes a simplifying assumption that the features are conditionally independent of each other given the class label.

The Naive Bayes algorithm can be used for binary as well as multi-class classification problems. It is commonly used in text classification tasks, such as spam filtering or sentiment analysis, but it can also be used in other applications where there are multiple classes and multiple features.

Performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values.



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

It is extremely useful for measuring Recall, Precision, Specificity, Accuracy, and most importantly AUC-ROC curves.

True Positive:

Interpretation: You predicted positive and it’s true.

True Negative:

Interpretation: You predicted negative and it’s true.

False Positive: (Type 1 Error)

Interpretation: You predicted positive and it’s false.

False Negative: (Type 2 Error)

Interpretation: You predicted negative and it’s false.

Recall

$$\textit{Recall} = \frac{TP}{TP + FN}$$

The above equation can be explained by saying, from all the positive classes, how many we predicted correctly.

Recall should be as high as possible.

Precision

$$\textit{Precision} = \frac{TP}{TP + FP}$$

The above equation can be explained by saying, from all the classes we have predicted as positive, how many are actually positive.

Precision should be as high as possible.

Accuracy

how many of them we have predicted correctly.

Accuracy should be as high as possible.

F-measure

$$\textbf{\textit{F - measure}} = \frac{2*\text{Recall}*\text{Precision}}{\text{Recall} + \text{Precision}}$$

It is difficult to compare two models with low precision and high recall or vice versa. So to make them comparable, we use F-Score. F-score helps to measure Recall and Precision at the same time. It uses Harmonic Mean in place of Arithmetic Mean by punishing the extreme values more.

Conclusion: Hence we have thoroughly studied how to perform the following operations using Python on created dataset (e.g. data.csv / Dictionary)