Assignment 6:

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 classifiers are a collection of classification algorithms based on **Bayes' Theorem**. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

Bayes' Theorem

Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes' theorem is stated mathematically as the following equation:

where A and B are events and $P(B) \neq 0$.

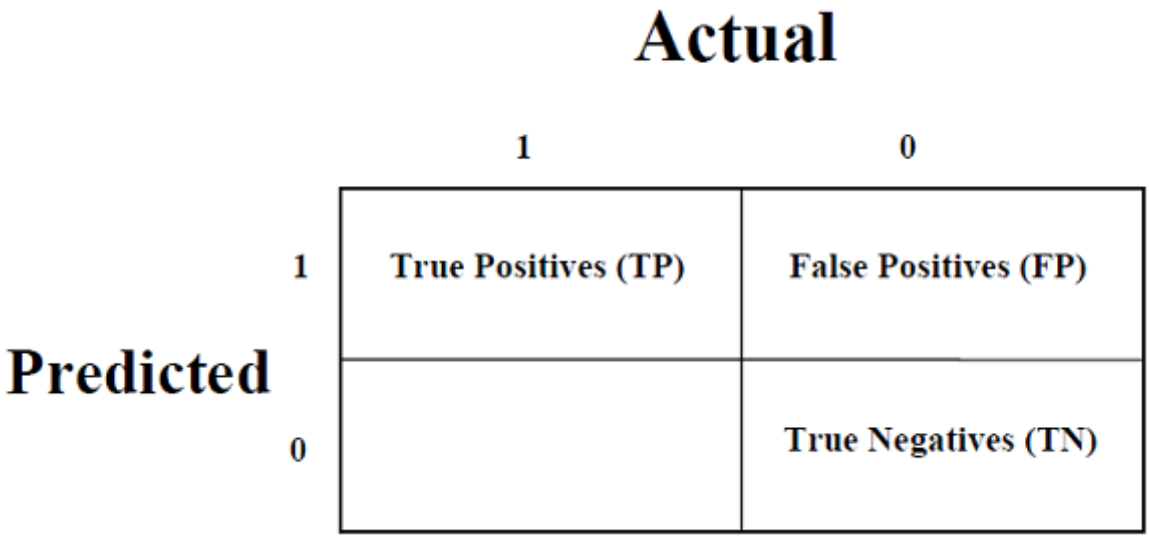
Basically, we are trying to find probability of event A, given the event B is true. Event B is also termed as evidence.

P(A) is the priori of A (the prior probability, i.e. Probability of event before evidence is seen). The evidence is an attribute value of an unknown instance(here, it is event B).

P(A|B) is a posteriori probability of B, i.e. probability of event after evidence is seen.

Confusion Matrix:

It is the easiest way to measure the performance of a classification problem where the output can be of two or more type of classes. A confusion matrix is nothing but a table with two dimensions viz. "Actual" and "Predicted" and furthermore, both the dimensions have "True Positives (TP)", "True Negatives (TN)", "False Positives (FP)", "False Negatives (FN)" as shown below



- True Positives (TP) − It is the case when both actual class & predicted class of data point is 1.
- True Negatives (TN) − It is the case when both actual class & predicted class of data point is 0.
- False Positives (FP) − It is the case when actual class of data point is 0 & predicted class of data point is 1.
- False Negatives (FN) − It is the case when actual class of data point is 1 & predicted class of data point is 0

- It is most common performance metric for classification algorithms. It may be defined as the number of correct predictions made as a ratio of all predictions made. We can easily calculate it by confusion matrix with the help of following formula −
- Accuracy=TP+TN/TP+FP+FN+TN
- We can use accuracy_score function of sklearn.metrics to compute accuracy of our classification model.
- Classification Report

This report consists of the scores of Precisions, Recall, F1 and Support. They are explained as follows

- Precision
- Precision, used in document retrievals, may be defined as the number of correct documents returned by our ML model. We can easily calculate it by confusion matrix with the help of following formula −
- Precision=TP/TP+FP

- Recall or Sensitivity
- Recall may be defined as the number of positives returned by our ML model. We can easily calculate it by confusion matrix with the help of following formula −
- Recall=TP/TP+FN

- Specificity
- Specificity, in contrast to recall, may be defined as the number of negatives returned by our ML model. We can easily calculate it by confusion matrix with the help of following formula
- Specificity=TN/TN+FP

Conclusion: Thus we have implemented python program for Naïve Bayes Classifier for Iris data Set. Accuracy of our model was found to be: