DS&BDL
Assignment 6

Title: Data Analytics III

Aim

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.

Objective:

Students are able to learn:

1. How to calculate the probabilities required by the Naive Bayes algorithm.
2. How to implement the Naive Bayes algorithm from scratch.
3. How to apply Naive Bayes to a real-world predictive modeling problem.

Software/Hardware Requirements: Python/R, on iris.csv dataset, Linux Operating System.

Theory:

Naive Bayes

Bayes' Theorem provides a way that we can calculate the probability of a piece of data belonging to a given class, given our prior knowledge. Bayes' Theorem is stated as:

Classification is a predictive modeling problem that involves assigning a label to a given input data sample.

The problem of classification predictive modeling can be framed as calculating the conditional probability of a class label given a data sample, for example:

P(class|data) = (P(data|class) * P(class)) / P(data)

Where P(class|data) is the probability of class given the provided data.

This calculation can be performed for each class in the problem and the class that is assigned the largest probability can be selected and assigned to the input data.

Naive Bayes is a classification algorithm for binary (two-class) and multiclass classification problems. It is called Naive Bayes or idiot Bayes because the calculations of the probabilities for each class are simplified to make their calculations tractable.

Rather than attempting to calculate the probabilities of each attribute value, they are assumed to be conditionally independent given the class value.

For this implementation we will use the Iris Flower Species Dataset.

The Iris Flower Dataset involves predicting the flower species given measurements of iris flowers.

It is a multiclass classification problem. The number of observations for each class is balanced. There are 150 observations with 4 input variables and 1 output variable. The variable names are as follows:

- Sepal length in cm.
- Sepal width in cm.
- Petal length in cm.
- Petal width in cm.
- Class

Steps:

1. Load Database
2. Separate By Class.
3. Summarize Dataset.
4. Summarize Data By Class.
5. Gaussian Probability Density Function.
6. Class Probabilities.

The first step is to load the dataset and convert the loaded data to numbers that we can use with the mean and standard deviation calculations. For this we will use the helper function load_csv() to load the file, str_column_to_float() to convert string numbers to floats and str_column_to_int() to convert the class column to integer values.

We will evaluate the algorithm using k-fold cross-validation with 5 folds. This means that 150/5=30 records will be in each fold. We will use the helper functions evaluate_algorithm() to evaluate the algorithm with cross-validation and accuracy_metric() to calculate the accuracy of predictions.

A new function named predict() was developed to manage the calculation of the probabilities of a new row belonging to each class and selecting the class with the largest probability value.

Another new function named naive_bayes() was developed to manage the application of the Naive Bayes algorithm, first learning the statistics from a training dataset and using them to make predictions for a test dataset.

Confusion Matrix Definition

A confusion matrix is used to judge the performance of a classifier on the test dataset for which we already know the actual values. Confusion matrix is also termed as Error matrix. It consists of a count of correct and incorrect values broken down by each class. It not only tells us the error made by classifier but also tells us what type of error the classifier made. So, we can say that a confusion matrix is a performance measurement technique of a classifier model where output can be two classes or more. It is a table with four different groups of true and predicted values.

Terminologies in Confusion Matrix

The confusion matrix shows us how our classifier gets confused while predicting. In a confusion matrix we have four important terms which are:

1. **True Positive (TP)-** Both actual and predicted values are Positive.
2. **True Negative (TN)-** Both actual and predicted values are Negative.
3. **False Positive (FP)-** The actual value is negative but we predicted it as positive.
4. **False Negative (FN)-** The actual value is positive but we predicted it as negative.

Performance Metrics

Confusion matrix not only used for finding the errors in prediction but is also useful to find some important performance metrics like Accuracy, Recall, Precision, F-measure. We will discuss these terms one by one.

Accuracy

As the name suggests, the value of this metric suggests the accuracy of our classifier in predicting results.

It is defined as:

Accuracy = (TP + TN) / (TP + TN + FP + FN)
A 99% accuracy can be good, average, poor or dreadful depending upon the problem.

Precision

Precision is the measure of all actual positives out of all predicted positive values.

It is defined as:

Precision = TP / (TP + FP)

Recall

Recall is the measure of positive values that are predicted correctly out of all actual positive values.

It is defined as:

Recall = TP / (TP + FN)
High Value of Recall specifies that the class is correctly known (because of a small number of False Negative).

F-measure

It is hard to compare classification models which have low precision and high recall or vice versa. So, for comparing the two classifier models we use F-measure. F-score helps to find the metrics of Recall and Precision in the same interval. Harmonic Mean is used instead of Arithmetic Mean.

F-measure is defined as:

F-measure = 2 * Recall * Precision / (Recall + Precision)
The F-Measure is always closer to the Precision or Recall, whichever has a smaller value

Conclusion: In this assignment, we covered how Naïve Bayes theorem used to solve classification problem for iris flower dataset and what is confusion matrix, its need, and how to derive it in Python and R.