Assignment No: 6

Aim/Problem Statement:- Data Analytics III

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

Theory:-

Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems. It is mainly used in *text classification* that includes a high-dimensional training dataset. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. **It is a probabilistic classifier, which means it predicts on the basis of the probability of an object**. Some popular examples of Naïve Bayes Algorithm are **spam filtration, Sentimental analysis, and classifying articles**.

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

- **Naïve**: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- **Bayes**: It is called Bayes because it depends on the principle of Bayes' Theorem.

Bayes' Theorem:

- Bayes' theorem is also known as **Bayes' Rule** or **Bayes' law**, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.
- The formula for Bayes' theorem is given as:

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

Where,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

Advantages of Naïve Bayes Classifier:

- Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
- It can be used for Binary as well as Multi-class Classifications.

- It performs well in Multi-class predictions as compared to the other Algorithms.
- It is the most popular choice for **text classification problems**.

Disadvantages of Naïve Bayes Classifier:

- Naive Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.

Applications of Naïve Bayes Classifier:

- It is used for **Credit Scoring**.
- It is used in **medical data classification**.
- It can be used in **real-time predictions** because Naïve Bayes Classifier is an eager learner.
- It is used in Text classification such as **Spam filtering** and **Sentiment analysis**.

Types of Naïve Bayes Model:

There are three types of Naive Bayes Model, which are given below:

- **Gaussian**: The Gaussian model assumes that features follow a normal distribution. This means if predictors take continuous values instead of discrete, then the model assumes that these values are sampled from the Gaussian distribution.
- **Multinomial**: The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed. It is primarily used for document classification problems, it means a particular document belongs to which category such as Sports, Politics, education, etc. The classifier uses the frequency of words for the predictors.
- **Bernoulli**: The Bernoulli classifier works similar to the Multinomial classifier, but the predictor variables are the independent Booleans variables. Such as if a particular word is present or not in a document. This model is also famous for document classification tasks.

Algorithm:-

Step 1: Download the data set of Iris
(https://www.kaggle.com/uciml/iris)

Step 2: Importing Libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Step 3: Assign the 4 independent variables to *X* and the dependent variable 'species' to *Y*.
The first 5 rows of the dataset are displayed

```
X = dataset.iloc[:,:4].values
y = dataset['species'].valuesdataset.head(5)
```

Step 4: Splitting the dataset into the Training set and Test set

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)
```

Step 5: Feature Scaling

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
```

```
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Step 6: Training the Naive Bayes Classification model on the Training Set

```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)
```

Step 7: Predicting the Test set results

```
y_pred = classifier.predict(X_test)
y_pred
```

Step 8: Confusion Matrix and Accuracy

```
from sklearn.metrics import confusion_matrix,classification_report
cm = confusion_matrix(y_test, y_pred)
from sklearn.metrics import accuracy_score
print ("Accuracy : ", accuracy_score(y_test, y_pred))
print(cm)
print("Error rate:",(1-accuracy_score(y_test, y_pred)))
```

Step 9: Compute Precision and Recall

```
cl_report=classification_report(y_test,y_pred)
print(cl_report)
```

Step 10: #Comparing the Real Values with Predicted Values

```
df = pd.DataFrame({'Real Values':y_test, 'Predicted Values':y_pred})
df
```

Input: Iris Dataset

Output: Confusion matrix, Accuracy, Error rate, Precision, Recall on the given dataset.

Conclusion: Implemented successfully Simple Naïve Bayes classification algorithm using Python on iris.csv dataset

Questions:

1) What is confusion matrix
2) How to calculate Accuracy, precision and recall
3) Explain applications of naïve Bays classification algorithm
4) State advantages and limitations of Naïve Bays algorithm