Name: Girish Nandanwar	Course: 22CT744 – Lab. Machine Learning
Roll No: A-56	Department: Computer Technology

Practical No.2

Aim: Implement the Naive Bayes Classifier and Analyse Evaluation Metrics

Theory:

Naive Bayes Classifier

Naive Bayes is a probabilistic machine learning algorithm based on Bayes' Theorem with the "naive" assumption that all features are conditionally independent. It is widely used for classification tasks such as spam detection, text classification, and medical diagnosis.

Bayes' Theorem:
(P(y|X) = \frac{P(X|y) \cdot P(y)}{P(X)})

Where:

- (P(y|X)): Posterior probability of class (y) given features (X)
- (P(X|y)): Likelihood of features (X) given class (y)
- (P(y)): Prior probability of class (y)
- (P(X)): Evidence (probability of features)

Gaussian Naive Bayes assumes that continuous features follow a Gaussian (normal) distribution:
(P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right))

Evaluation Metrics

Accuracy – Ratio of correctly predicted observations to total observations
(\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN})

Precision – Ratio of correctly predicted positive observations to total predicted positives
(\text{Precision} = \frac{TP}{TP + FP})

Recall (Sensitivity) – Ratio of correctly predicted positive observations to actual positives
(\text{Recall} = \frac{TP}{TP + FN})

F1-Score – Harmonic mean of precision and recall

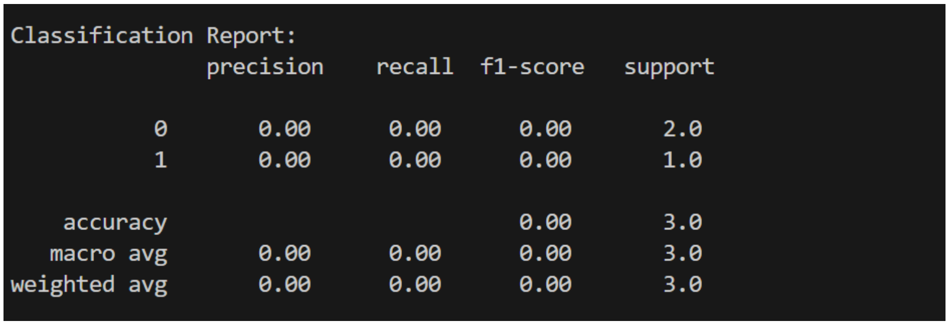
(\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}})

Code:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Dataset
data = {
   "Country": ["France", "Spain", "Germany", "Spain", "Germany", "France", "Spain", "France",
"Germany", "France"],
   "Age": [44, 27, 30, 38, 40, 35, None, 48, 50, 37],
   "Salary": [72000, 48000, 54000, 61000, None, 58000, 52000, 79000, 83000, 67000],
   "Purchased": ["No", "Yes", "No", "No", "Yes", "Yes", "No", "Yes", "No", "Yes"]
}

df = pd.DataFrame(data)

# Handle missing values
df["Age"].fillna(df["Age"].mean(), inplace=True)
df["Salary"].fillna(df["Salary"].mean(), inplace=True)

# Encode categorical data
le_country = LabelEncoder()
df["Country"] = le_country.fit_transform(df["Country"])

le_purchase = LabelEncoder()
df["Purchased"] = le_purchase.fit_transform(df["Purchased"])

# Split features and target
X = df[["Country", "Age", "Salary"]]
y = df["Purchased"]

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)

# Naive Bayes model
nb = GaussianNB()
nb.fit(X_train, y_train)
```

```
y_pred = nb.predict(X_test)

# Evaluation metrics
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

Result:



```
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       2.0
           1       0.00      0.00      0.00       1.0

    accuracy                           0.00       3.0
   macro avg       0.00      0.00      0.00       3.0
weighted avg       0.00      0.00      0.00       3.0
```

Conclusion:

This practical helped us learn how to clean, analyze, and visualize data using Python libraries. These steps are important for building accurate machine learning mode.