EXPERIMENT – 19

Implementation of Naive Bayes classification for Bank Loan prediction

CODE :

```
# Bank Loan Prediction using Naive Bayes

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, classification_report

# Sample dataset
data = {
    "Age": [25, 45, 35, 50, 23, 40, 60, 48],
    "Income": [30000, 80000, 50000, 90000, 28000, 65000, 120000, 70000],
    "Credit_History": ["Good", "Good", "Bad", "Good", "Bad", "Bad", "Good", "Good"],
    "Loan_Approved": ["Yes", "Yes", "No", "Yes", "No", "No", "Yes", "Yes"]
}

df = pd.DataFrame(data)

# Encode categorical variables
le_credit = LabelEncoder()
df["Credit_History"] = le_credit.fit_transform(df["Credit_History"])

le_target = LabelEncoder()
df["Loan_Approved"] = le_target.fit_transform(df["Loan_Approved"])

# Features and target
X = df[["Age", "Income", "Credit_History"]]
y = df["Loan_Approved"]

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=42
)

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

```
# Prediction
y_pred = model.predict(X_test)

# Evaluation (zero_division=0 prevents warnings)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n",
      classification_report(y_test, y_pred, target_names=le_target.classes_, zero_division=0))

# Predict for a new customer
new_customer = pd.DataFrame(
    [[30, 60000, "Good"]],
    columns=["Age", "Income", "Credit_History"]
)

# Encode credit history
new_customer["Credit_History"] = le_credit.transform(new_customer["Credit_History"])

prediction = model.predict(new_customer)
print("Loan Approval Prediction:", le_target.inverse_transform(prediction)[0])
```

OUTPUT :