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

# Sample bank loan dataset
data = {
    "Age": [25, 35, 45, 23, 52, 40, 28, 60],
    "Income": [30000, 60000, 80000, 25000, 90000, 70000, 35000,
100000],
    "LoanAmount": [5000, 15000, 20000, 4000, 25000, 18000, 7000,
30000],
    "CreditScore": [650, 720, 750, 600, 780, 740, 680, 800],
    "LoanApproved": ["No", "Yes", "Yes", "No", "Yes", "Yes", "No",
"Yes"]
}

df = pd.DataFrame(data)
df
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 8,\n  \"fields\": [\n
{\n      \"column\": \"Age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 13,\n        \"min\": 23,\n
\"max\": 60,\n        \"num_unique_values\": 8,\n        \"samples\":
[\n          35,\n          40,\n          25\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Income\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 28629,\n
\"min\": 25000,\n        \"max\": 100000,\n
\"num_unique_values\": 8,\n        \"samples\": [\n          60000,\n
70000,\n          30000\n        ],\n        \"semantic_type\": \"\",\
n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"LoanAmount\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 9576,\n        \"min\":
4000,\n        \"max\": 30000,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          15000,\n          18000,\n          5000\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"CreditScore\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
67,\n        \"min\": 600,\n        \"max\": 800,\n
\"num_unique_values\": 8,\n        \"samples\": [\n          720,\n
740,\n          650\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"LoanApproved\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"Yes\",\n          \"No\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```

```
# Convert target labels to numerical values
df["LoanApproved"] = df["LoanApproved"].map({"No": 0, "Yes": 1})

X = df[["Age", "Income", "LoanAmount", "CreditScore"]]
y = df["LoanApproved"]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=42
)

model = GaussianNB()
model.fit(X_train, y_train)
```

```
GaussianNB()
```

```
y_pred = model.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n",
      classification_report(y_test, y_pred, zero_division=0))
```

```
Accuracy: 1.0

Classification Report:
               precision    recall  f1-score   support

           1       1.00      1.00      1.00         2

    accuracy                           1.00         2
   macro avg       1.00      1.00      1.00         2
weighted avg       1.00      1.00      1.00         2
```

```
# New customer details
new_customer = pd.DataFrame(
    [[30, 50000, 10000, 700]],
    columns=["Age", "Income", "LoanAmount", "CreditScore"]
)

prediction = model.predict(new_customer)

if prediction[0] == 1:
    print("Loan Status: Approved")
else:
    print("Loan Status: Not Approved")
```

```
Loan Status: Not Approved
```