CSL 325
Artificial Intelligence Lab

Department of Computer Science
Bahria University, Islamabad

Ammar Jamil
01-134231-010

Instructor: Dr. Arshad Farhad

Lab # 12: Naïve Bayes

Lab Tasks:

Task 2: Implement Naive Bayes Classifier from Scratch.

Objective: Predict whether a customer will buy a product based on the given attributes.

Synthetic Dataset: "Online Purchase Prediction"

Attributes:

Age	Income	Student	Credit Rating	Buys Product (Label)
<=25	High	No	Fair	No
<=25	High	No	Excellent	No
26-35	High	No	Fair	Yes
>35	Medium	No	Fair	Yes
>35	Low	Yes	Fair	Yes
>35	Low	Yes	Excellent	No
26-35	Low	Yes	Excellent	Yes
<=25	Medium	No	Fair	No
<=25	Low	Yes	Fair	Yes
>35	Medium	Yes	Fair	Yes
<=25	Medium	Yes	Excellent	Yes
26-35	Medium	No	Excellent	Yes
26-35	High	Yes	Fair	Yes

Age	Income	Student	Credit Rating	Buys Product (Label)
>35	Medium	No	Excellent	No

Step 1: Calculate Priors

1. Compute P(Buys Product = Yes) and P(Buys Product = No) from the dataset.

Step 2: Calculate Likelihoods

For each attribute (Age, Income, Student, Credit Rating):

1. Calculate conditional probabilities, e.g.,
    - P(Age <=25 | Buys Product = Yes)
    - P(Income = High | Buys Product = No)
    - ... (and similarly for other attributes).

Step 3: Make a Prediction

Predict the class for the following new instances using the Naive Bayes formula:

1. `[Age = <=25, Income = Low, Student = Yes, Credit Rating = Fair]`
2. `[Age = >35, Income = High, Student = No, Credit Rating = Excellent]`

Step 4: Compare with `Sklearn`

Validate your results using `sklearn.naive_bayes.CategoricalNB.`

Note:

1. Use screenshot for each calculation step (results)
2. Submit the pdf

```
import pandas as pd
import numpy as np
from sklearn.naive_bayes import CategoricalNB
from sklearn.preprocessing import OrdinalEncoder


data = {
    'Age': ['<=25', '<=25', '26-35', '>35', '>35', '>35', '26-35', '<=25', '<=25', '>35', '<=25', '26-35',
'26-35', '>35'],
    'Income': ['High', 'High', 'High', 'Medium', 'Low', 'Low', 'Low', 'Medium', 'Low', 'Medium', 'Medium',
'Medium', 'High', 'Medium'],
    'Student': ['No', 'No', 'No', 'No', 'Yes', 'Yes', 'Yes', 'No', 'Yes', 'Yes', 'Yes', 'No', 'Yes', 'No'],
    'Credit_Rating': ['Fair', 'Excellent', 'Fair', 'Fair', 'Fair', 'Excellent', 'Excellent', 'Fair',
'Fair', 'Fair', 'Excellent', 'Excellent', 'Fair', 'Excellent'],
    'Buys_Product': ['No', 'No', 'Yes', 'Yes', 'Yes', 'No', 'Yes', 'No', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes',
'No']
}
```

```
df = pd.DataFrame(data)
X_test_raw = [
    ['<=25', 'Low', 'Yes', 'Fair'],
    ['>35', 'High', 'No', 'Excellent']
]
```

```
print(df)
print("\n")
```

```
def naive_bayes_scratch(df, instance):

    total_count = len(df)
    p_yes = len(df[df['Buys_Product'] == 'Yes']) / total_count
    p_no = len(df[df['Buys_Product'] == 'No']) / total_count
```

```
    df_yes = df[df['Buys_Product'] == 'Yes']
    df_no = df[df['Buys_Product'] == 'No']

    # Step 2: Calculate Conditional Probabilities (Likelihoods)
    # Instance attributes: Age, Income, Student, Credit
    prob_yes = p_yes
    prob_no = p_no

    features = ['Age', 'Income', 'Student', 'Credit_Rating']

    for i, feature in enumerate(features):
        val = instance[i]


        count_feat_yes = len(df_yes[df_yes[feature] == val])
        p_feat_given_yes = count_feat_yes / len(df_yes)
        prob_yes *= p_feat_given_yes


        count_feat_no = len(df_no[df_no[feature] == val])
        p_feat_given_no = count_feat_no / len(df_no)
        prob_no *= p_feat_given_no
```

```
    total_prob = prob_yes + prob_no
    norm_yes = prob_yes / total_prob
    norm_no = prob_no / total_prob

    prediction = "Yes" if prob_yes > prob_no else "No"
    return prediction, prob_yes, prob_no
```

```
for i, inst in enumerate(X_test_raw):
    pred, p_yes, p_no = naive_bayes_scratch(df, inst)
    print(f"Instance {i+1} {inst}: Prediction = {pred}")
    print(f"   Raw Probabilities -> Yes: {p_yes:.4f}, No: {p_no:.4f}")
```

```
Instance 1 ['<=25', 'Low', 'Yes', 'Fair']: Prediction = Yes
   Raw Probabilities =>  Yes: 0.0212, No: 0.0034
Instance 2 ['>35', 'High', 'No', 'Excellent']: Prediction = No
   Raw Probabilities =>  Yes: 0.0053, No: 0.0274
```

```
import pandas as pd
from sklearn.naive_bayes import CategoricalNB
from sklearn.preprocessing import OrdinalEncoder

# 1. Create the dataset
data = {
    'Age': ['<=25', '<=25', '26-35', '>35', '>35', '>35', '26-35', '<=25', '<=25', '>35', '<=25', '26-35',
'26-35', '>35'],
    'Income': ['High', 'High', 'High', 'Medium', 'Low', 'Low', 'Low', 'Medium', 'Low', 'Medium', 'Medium',
'Medium', 'High', 'Medium'],
    'Student': ['No', 'No', 'No', 'No', 'Yes', 'Yes', 'Yes', 'No', 'Yes', 'Yes', 'Yes', 'No', 'Yes', 'No'],
    'Credit Rating': ['Fair', 'Excellent', 'Fair', 'Fair', 'Fair', 'Excellent', 'Excellent', 'Fair',
'Fair', 'Fair', 'Excellent', 'Excellent', 'Fair', 'Excellent'],
    'Buys Product': ['No', 'No', 'Yes', 'Yes', 'Yes', 'No', 'Yes', 'No', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes',
'No']
}
```

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

```
encoder = OrdinalEncoder()
X_encoded = encoder.fit_transform(df[['Age', 'Income', 'Student', 'Credit Rating']])
y = df['Buys Product']
```

```
model = CategoricalNB(alpha=0)
model.fit(X_encoded, y)
```

```
new_instances = pd.DataFrame({
    'Age': ['<=25', '>35' ],
    'Income': ['Low', 'High'],
    'Student': ['Yes', 'No'],
    'Credit Rating': ['Fair', 'Excellent']
})
new_instances_encoded = encoder.transform(new_instances)
```

```
predictions = model.predict(new_instances_encoded)
probabilities = model.predict_proba(new_instances_encoded)
```

```
print(f"\nPrediction for Instance 1: {predictions[0]}")
print(f"Probabilities [P(No), P(Yes)]: {probabilities[0]}")
```

```
print(f"\nPrediction for Instance 2: {predictions[1]}")
print(f"Probabilities [P(No), P(Yes)]: {probabilities[1]}")
```

```
Prediction for Instance 1: Yes
Probabilities [P(No), P(Yes)]: [0.1394148 0.8605852]

Prediction for Instance 2: No
Probabilities [P(No), P(Yes)]: [0.83829237 0.16170763]

C:\Users\Ammar Jamil\Desktop\COURSE\6TH SEMESTER\AI LAB\ASSIGNMENTS\AILab13>
```