



NAIVE BAYES

```
In [6]: # Importing necessary libraries
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import CategoricalNB

# Creating the dataset
data = {
    'Cheap': ['Yes', 'Yes', 'No', 'Yes', 'No'],
    'Promo': ['Yes', 'No', 'Yes', 'Yes', 'No'],
    'Buy?': ['Yes', 'Yes', 'No', 'Yes', 'No']
}

# Creating a DataFrame
df = pd.DataFrame(data)
print(df)

# Encoding categorical variables
le_cheap = LabelEncoder()
le_promo = LabelEncoder()
le_buy = LabelEncoder()

df['Cheap'] = le_cheap.fit_transform(df['Cheap'])
df['Promo'] = le_promo.fit_transform(df['Promo'])
df['Buy?'] = le_buy.fit_transform(df['Buy?'])

# Splitting features and target
X = df[['Cheap', 'Promo']]
y = df['Buy?']
```

	Cheap	Promo	Buy?
0	Yes	Yes	Yes
1	Yes	No	Yes
2	No	Yes	No
3	Yes	Yes	Yes
4	No	No	No

```
In [7]: # Creating and training the Naive Bayes model
model = CategoricalNB()
model.fit(X, y)

# Making predictions
predictions = model.predict(X)
print(predictions)

# Adding predictions to the DataFrame
df['Predictions'] = predictions
print(df)
```

```
[1 1 0 1 0]
   Cheap  Promo  Buy?  Predictions
0      1      1      1           1
1      1      0      1           1
2      0      1      0           0
3      1      1      1           1
4      0      0      0           0
```

```
In [5]: # Converting encoded values back to original labels
df['Predicted_Buy?'] = le_buy.inverse_transform(predictions)
df['ActualBuy'] = le_buy.inverse_transform(y)

# Final output
print(df)
```

```
   Cheap  Promo  Buy?  Predictions  Predicted_Buy?  ActualBuy
0      1      1      1           1             Yes         Yes
1      1      0      1           1             Yes         Yes
2      0      1      0           0             No          No
3      1      1      1           1             Yes         Yes
4      0      0      0           0             No          No
```