Predicting Purchase Probability Using Naive Bayes Classifier

Here, we use a Naive Bayes Classifier to predict the probability of a purchase based on the features: visiting the homepage, having a coupon code, and being a repeat visitor.

1. Creating the dataset

First, we create a dataset directly within the code using a `pandas` DataFrame. The dataset contains user actions and their corresponding purchase decisions.

```
import pandas as pd

data = pd.DataFrame({
    'Feature': ['F1', 'F1', 'F2', 'F2', 'F1', 'F2', 'F3', 'F3', 'F1', 'F2', 'F2', 'F1'],
    'Action': ['Yes', 'Yes', 'Yes', 'Yes', 'No', 'No', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes'],
    'Sale': ['Yes', 'Yes', 'Yes', 'Yes', 'No', 'No', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes']
})

# Displaying the dataset
data
```

	Feature	Action	Sale
0	F1	Yes	Yes
1	F1	Yes	Yes
2	F2	Yes	Yes
3	F2	Yes	Yes
4	F1	No	No
5	F2	No	No
6	F3	No	No
7	F3	Yes	Yes
8	F1	No	No
9	F2	Yes	Yes
10	F2	No	No
11	F1	Yes	Yes

Next steps: Generate code with `data` View recommended plots New interactive sheet

2. Creating a frequency table

Next, we create a frequency table that summarises the counts of actions (features) corresponding to each sale outcome (`Yes` or `No`). This table will help us understand how many users took each action and whether it led to a purchase.

```
frequency_table = data.groupby(['Feature', 'Sale']).size().unstack(fill_value=0)
frequency_table
```

Sale	No	Yes
Feature		
F1	2	3
F2	2	3
F3	1	1

Next steps: Generate code with frequency_table | View recommended plots | New interactive sheet

3. Creating the likelihood table

Based on the frequency table, we compute the likelihood table, which contains the conditional probabilities of each feature given the sale outcome. This table will be used to estimate the probability of a sale based on the features.

```
likelihood_table = frequency_table.div(frequency_table.sum(axis=1), axis=0)
likelihood_table
```

Sale	No	Yes
Feature		
F1	0.4	0.6
F2	0.4	0.6
F3	0.5	0.5

Next steps: Generate code with likelihood_table | View recommended plots | New interactive sheet

4. Applying Bayes' theorem

Finally, we define prior probabilities for purchases (`Yes` and `No`) based on the overall dataset. We then calculate the probabilities of making a purchase given the presence of specific features using Bayes' theorem. The probabilities are normalised to ensure they sum to 1.

```
# Calculating prior probabilities
prior_yes = len(data[data['Sale'] == 'Yes']) / len(data)
prior_no = len(data[data['Sale'] == 'No']) / len(data)

# Defining the user input (features)
user_input = {
    'F1': 1,  # Visits home page
    'F2': 1,  # Has a coupon code
    'F3': 1   # Is a repeat visitor
}

# Calculating probabilities using Bayes' theorem
def calculate_purchase_probability(user_input, likelihood_table, prior_yes, prior_no):
    prob_yes = prior_yes
    prob_no = prior_no

    for feature, value in user_input.items():
        if value == 1:  # If feature is present
            prob_yes *= likelihood_table.loc[feature, 'Yes']  # Likelihood of purchase given feature
            prob_no *= likelihood_table.loc[feature, 'No']    # Likelihood of no purchase given feature

    return prob_yes, prob_no

prob_yes, prob_no = calculate_purchase_probability(user_input, likelihood_table, prior_yes, prior_no)

# Normalizing the probabilities
total_prob = prob_yes + prob_no
prob_yes_normalized = prob_yes / total_prob
prob_no_normalized = prob_no / total_prob
```

5. Conclusion

Based on the input features (visiting the home page, having a coupon code, and being a repeat visitor), we calculated the probabilities of making a purchase. The following results indicate the likelihood of a user buying something given these conditions.

```python
print("Prior probabilities:")
print(f"P(Sale = Yes) = {prior_yes:.4f}")
print(f"P(Sale = No) = {prior_no:.4f}")

print(f"\nP(Buy = Yes | Features) = {prob_yes_normalized:.4f}")
print(f"P(Buy = No | Features) = {prob_no_normalized:.4f}")

if prob_yes_normalized > prob_no_normalized:
  label = 'Yes'
else:
  label = 'No'
print(f"The predicted label for the input is: {label}")
```

```
Prior probabilities:
P(Sale = Yes) = 0.5833
P(Sale = No) = 0.4167

P(Buy = Yes | Features) = 0.7590
P(Buy = No | Features) = 0.2410
The predicted label for the input is: Yes
```