Praticle 7 : Implementation of Naive Bayes Classification Algorithms

In [27]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn.naive_bayes import CategoricalNB
```

In [28]:
```
# Generate synthetic dataset
X, y = make_blobs(n_samples=700, centers=4, random_state=42)
```

In [29]:
```
# Shift data to make all values non-negative
X_min = np.min(X)
X_shifted = X - X_min
```

In [30]:
```
# Creating a pandas DataFrame
nbflu = pd.DataFrame(X_shifted, columns=['x1', 'x2'])
nbflu['y'] = y
print(nbflu)
```

```
           x1         x2  y
0    14.633560  13.928388  1
1     4.474987   5.615441  2
2     1.151316  20.465349  3
3     0.560185  19.089828  3
4    17.199910  13.619839  1
..         ...        ... ..
695   1.564293  19.656543  3
696   3.885655   6.324588  2
697   2.856557  17.961496  3
698   3.957422   6.393296  2
699  15.999374  12.821994  1

[700 rows x 3 columns]
```

In [31]:
```
# Encoding the categorical variable
le = preprocessing.LabelEncoder()
y_encoded = le.fit_transform(y)
model = CategoricalNB()
```

In [32]:
```
# Train the model using the training sets
model.fit(X_shifted, y_encoded)

# Predict Output
# ['Y','N','Mild','Y']
predicted = model.predict([[1, 0]])
predicted_proba = model.predict_proba([[1, 0]])
print("Predicted Value:", predicted)
print("Predicted Probabilities:", predicted_proba)
```

```
Predicted Value: [3]
Predicted Probabilities: [[0.0212766  0.0212766  0.0212766  0.93617021]]
```

In [33]:
```
# Plotting the data
plt.figure(figsize=(8, 6))
plt.scatter(X_shifted[:, 0], X_shifted[:, 1], c=y_encoded, cmap='viridis'
plt.xlabel('x1')
plt.ylabel('x2')
plt.title('Categorical Naive Bayes')

plt.show()
```