

```
In [31]: data.head()
```

```
Out[31]:
```

|   | age | workclass        | fnlwgt | education | education-num | marital-status     | occupation        | relationship  | race  | sex    | capital-gain | capital-loss | hours-per-week | native-country | income |
|---|-----|------------------|--------|-----------|---------------|--------------------|-------------------|---------------|-------|--------|--------------|--------------|----------------|----------------|--------|
| 0 | 39  | State-gov        | 77516  | Bachelors | 13            | Never-married      | Adm-clerical      | Not-in-family | White | Male   | 2174         | 0            | 40             | United-States  | <=50K  |
| 1 | 50  | Self-emp-not-inc | 83311  | Bachelors | 13            | Married-civ-spouse | Exec-managerial   | Husband       | White | Male   | 0            | 0            | 13             | United-States  | <=50K  |
| 2 | 38  | Private          | 215646 | HS-grad   | 9             | Divorced           | Handlers-cleaners | Not-in-family | White | Male   | 0            | 0            | 40             | United-States  | <=50K  |
| 3 | 53  | Private          | 234721 | 11th      | 7             | Married-civ-spouse | Handlers-cleaners | Husband       | Black | Male   | 0            | 0            | 40             | United-States  | <=50K  |
| 4 | 28  | Private          | 338409 | Bachelors | 13            | Married-civ-spouse | Prof-specialty    | Wife          | Black | Female | 0            | 0            | 40             | Cuba           | <=50K  |

Train the Naive Bayes classifier:

```
In [17]: classifier = GaussianNB()  
classifier.fit(X_train, y_train)
```

```
Out[17]:  
GaussianNB()  
GaussianNB()
```

Calculate Accuracy:

```
In [19]: accuracy = accuracy_score(y_test, y_pred)  
print("Accuracy:", accuracy)  
Accuracy: 0.8004913501893746
```

Compute confusion matrix:

```
In [20]: confusion = confusion_matrix(y_test, y_pred)  
print("Confusion Matrix:")  
print(confusion)  
Confusion Matrix:  
[[7081 334]  
 [1615 739]]
```

Compute sensitivity (true positive rate):

```
In [23]: sensitivity = tp / (tp + fn)  
print("Sensitivity:", sensitivity)  
Sensitivity: 0.3139337298215803
```

Compute specificity (true negative rate):

```
In [25]: specificity = tn / (tn + fp)  
print("Specificity:", specificity)  
Specificity: 0.954956169925826
```

Compute the posterior probability of making over 50K a year:

```
In [28]: y_prob = classifier.predict_proba(X_test)
        pos_prob = y_prob[:, 1]

In [30]: for prob in pos_prob:
        print("Posterior Probability of making over 50K a year:", prob)
```

Posterior Probability of making over 50K a year: 0.00858779393896831  
Posterior Probability of making over 50K a year: 0.021380593699508094  
Posterior Probability of making over 50K a year: 0.011819324809722764  
Posterior Probability of making over 50K a year: 0.03396414108955507  
Posterior Probability of making over 50K a year: 0.004270080675351556  
Posterior Probability of making over 50K a year: 0.9712272466160277  
Posterior Probability of making over 50K a year: 0.0016911059071351828  
Posterior Probability of making over 50K a year: 0.06365624531911285  
Posterior Probability of making over 50K a year: 0.0037895163857533934  
Posterior Probability of making over 50K a year: 0.008442412852003388  
Posterior Probability of making over 50K a year: 0.0017256406405623478  
Posterior Probability of making over 50K a year: 0.0018875212159514794  
Posterior Probability of making over 50K a year: 0.012178200650685426  
Posterior Probability of making over 50K a year: 0.04765791861616365  
Posterior Probability of making over 50K a year: 0.0370044419231842  
Posterior Probability of making over 50K a year: 0.00047197480040402843  
Posterior Probability of making over 50K a year: 0.0066987174066986165  
Posterior Probability of making over 50K a year: 0.0049774048109906665  
Posterior Probability of making over 50K a year: 0.999999937611008

Activate Window  
Go to Settings to a