

Naïve Bayes

Thursday, June 29, 2023 2:17 PM

↖ Indep ↗

Talks for more than 100 min? (TT >= 100)	Gender	Response
y	male	not bought
n	male	not bought
n	female	not bought
n	female	not bought
n	male	not bought
n	male	not bought
y	male	bought
y	female	bought
n	female	bought
y	female	bought

$$P(B | TT \geq 100, m)$$

$$= \frac{P(TT \geq 100 \cap M | B) P(B)}{P(TT \geq 100 \cap M | B) P(B) + P(TT \geq 100 \cap M | NB) P(NB)}$$

$$= \frac{P(TT \geq 100 | B) P(M | B) P(B)}{P(TT \geq 100 | B) P(M | B) P(B) + P(TT \geq 100 | NB) P(M | NB) P(NB)}$$

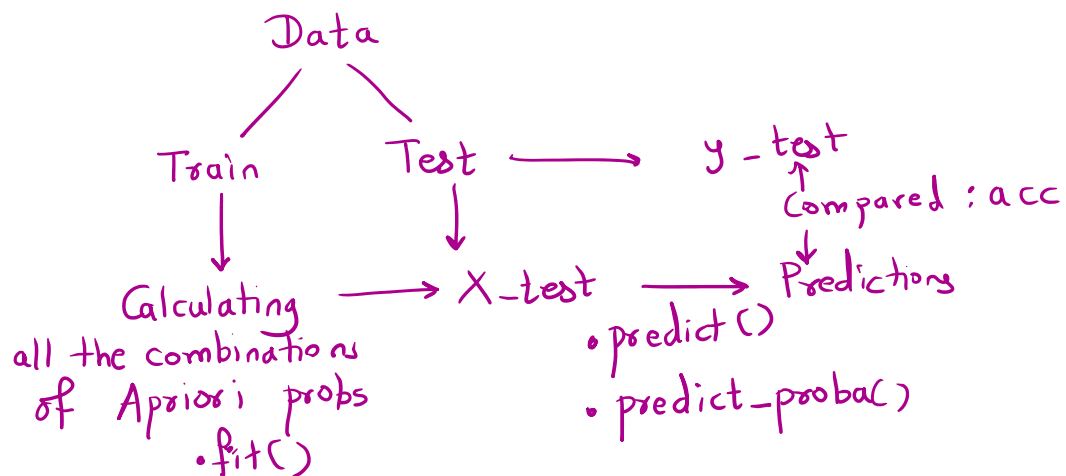
↖ Apriori Probabilities

$$= \frac{\frac{3}{4} \times \frac{1}{4} \times \frac{4}{10}}{\frac{3}{4} \times \frac{1}{4} \times \frac{4}{10} + \frac{1}{6} \times \frac{4}{6} \times \frac{6}{10}} = 0.529$$

↓ Posterior Probability

$$P(B | TT \geq 100 \cap F) = \frac{P(TT \geq 100 | B) P(F | B) P(B)}{P(TT \geq 100 | B) P(F | B) P(B) + P(TT \geq 100 | NB) P(F | NB) P(NB)}$$

$$= \frac{\frac{3}{4} \times \frac{3}{4} \times \frac{4}{10}}{\frac{3}{4} \times \frac{3}{4} \times \frac{4}{10} + \frac{1}{6} \times \frac{2}{6} \times \frac{6}{10}} = 0.87$$



Discrete NB

Kernel NB

Categorical Features	Numerical Features
Apriori Probabilities are calculated based on counts	Apriori Probabilities are calculated based on the function of Normal Distribution
Bayes Formula	Bayes Formula

[Stratification | Why to Stratify? | stratify=y option](#)

```

1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3
4 cancer = pd.read_csv("breastcancer.csv")
5 print(cancer.columns)
6
7 y = cancer['class']
8 X = cancer.drop('class', 'axis=1')
9
10 y_train, y_test, X_train, X_test = train_test_split(X, y,
11                                                    test_size=0.2,
12                                                    random_state=0)
13
14 y_train.value_counts(normalize=True)*100
15 y_test.value_counts(normalize=True)*100
16

```

The screenshot shows a Jupyter Notebook interface. The left pane contains the Python code for loading a dataset, splitting it into training and testing sets while maintaining class distribution (stratification), and calculating the normalized value counts for both sets. The right pane shows the output of the value_counts function, displaying the distribution of the 'class' variable for both training and testing data, confirming that the stratification was successful as the distributions are very similar.