

Naive Bayes



**UNIVERSIDAD
CATÓLICA**
DE CÓRDOBA
JESUITAS

Dr. Francisco Arduh
2023

Probabilidad condicional

¿Cuál es la probabilidad que salga un 6?



Probabilidad condicional

¿Cuál es la probabilidad que salga un 6?

$$P(seis) = \frac{1}{6}$$



Probabilidad condicional

¿Cuál es la probabilidad que salga un 6?

$$P(seis) = \frac{1}{6}$$

¿Cuál es la probabilidad que salga un 6 si el número que salió es par?



Probabilidad condicional

¿Cuál es la probabilidad que salga un 6?

$$P(seis) = \frac{1}{6}$$

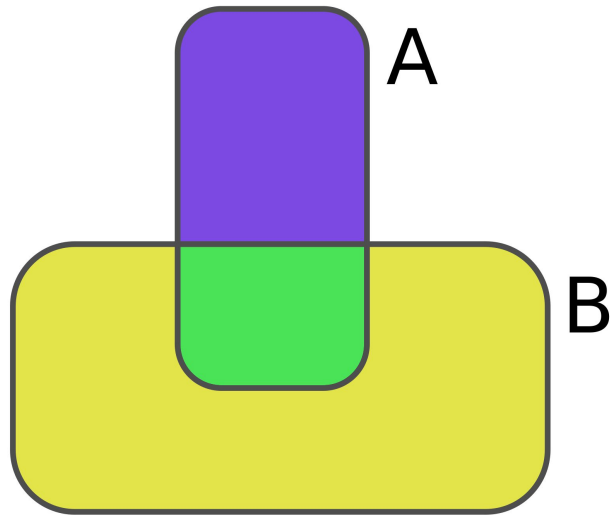
¿Cuál es la probabilidad que salga un 6 si el número que salió es par?

$$P(seis|par) = \frac{1}{3}$$



Probabilidad condicional

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

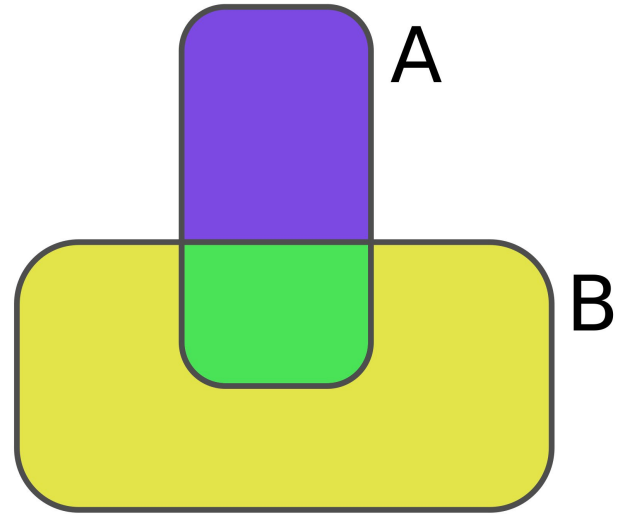


Probabilidad condicional

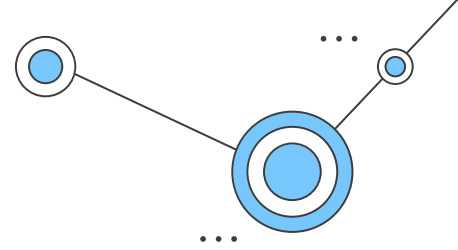
$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Esto es válido para múltiples ocurrencias de eventos:

$$P(y|x_1, \dots, x_n) = \frac{P(y, x_1, \dots, x_n)}{P(x_1, \dots, x_n)}$$



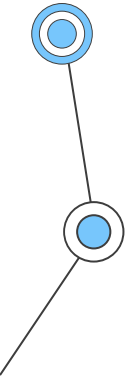
Teorema de Bayes



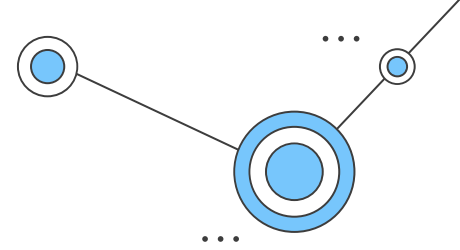
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Al igual que lo anterior es válido para múltiples ocurrencias de eventos:

$$P(y \mid x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n \mid y)}{P(x_1, \dots, x_n)}$$



Independencia

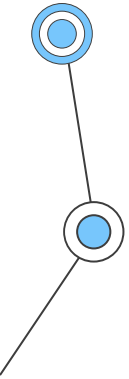


Decimos que los eventos A y B son independientes si:

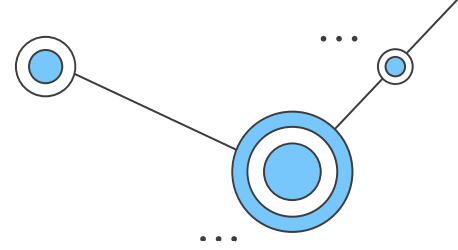
$$P(A|B) = P(A)$$

Si A y B son independientes, también se tiene que:

$$P(A, B|C) = P(A|C)P(B|C)$$



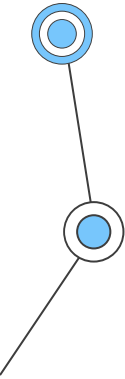
Volviendo a lo anterior



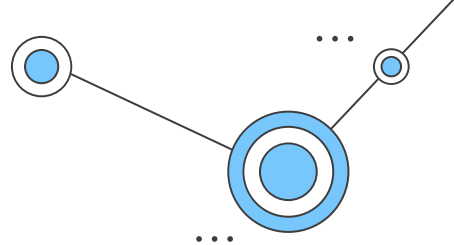
$$P(y \mid x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n \mid y)}{P(x_1, \dots, x_n)}$$

Si \mathbf{x} son independientes se puede escribir como

$$P(y \mid x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^n P(x_i \mid y)}{P(x_1, \dots, x_n)}$$



Obteniendo la predicción



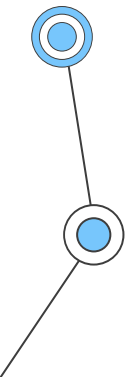
$$P(y \mid x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^n P(x_i \mid y)}{P(x_1, \dots, x_n)}$$

Notemos que sólo queremos obtener la clase con mayor probabilidad por lo que:

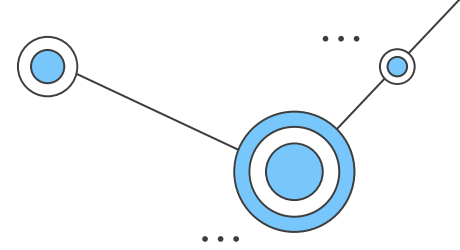
$$P(y \mid x_1, \dots, x_n) \propto P(y) \prod_{i=1}^n P(x_i \mid y)$$

\Downarrow

$$\hat{y} = \arg \max_y P(y) \prod_{i=1}^n P(x_i \mid y),$$



Veamos como funciona con un ejemplo



Se quiere predecir si lloverá o no, dadas las características meteorológicas.

La primera característica podría ser si está nublado o soleado, la segunda si la humedad es alta o baja, y la tercera si la temperatura es alta, media o baja.

```
<Nublado, H_Alta, T_Baja> -> Llueve  
<Soleado, H_Baja, T_Media> -> No Llueve  
<Nublado, H_Baja, T_Alta> -> No Llueve
```

Lo que se quiere obtener es:

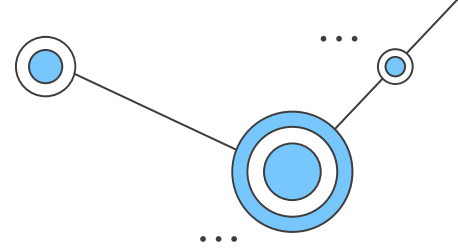
$$P(\text{Llueve} | \text{Nublado}, H_Alta, T_Baja)$$

$$P(\text{NoLlueve} | \text{Nublado}, H_Alta, T_Baja)$$

A partir de la probabilidad más grande se puede saber si lloverá o no.



Veamos como funciona con un ejemplo



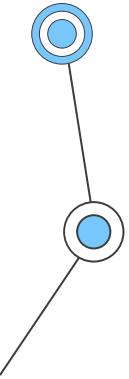
La probabilidad que llueva queda dada por:

$$P(\text{Llueve}|\text{Nublado}, \text{H_Baja}, \text{T_Baja}) \propto P(\text{Nublado}|\text{Llueve})P(\text{H_Baja}|\text{Llueve})P(\text{T_Baja}|\text{Llueve})P(\text{Llueve})$$

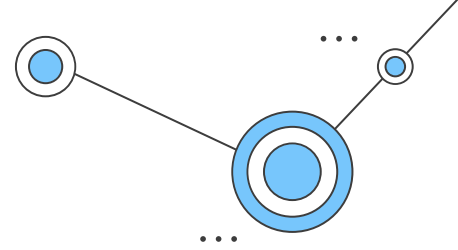
donde:

$$P(\text{Llueve}) = \frac{\text{\#instancias con lluvia}}{\text{\#total instancias}}$$

$$P(\text{Nublado}|\text{Llueve}) = \frac{\text{\#instancias nublado y con lluvia}}{\text{\#total instancias con lluvia}}$$



Naive Bayes en Scikit-learn

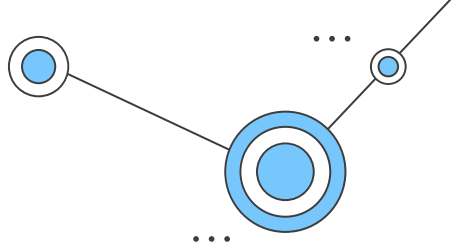


Algoritmo clásico de Naive Bayes (Se asume características discretas)

```
>>> import numpy as np
>>> rng = np.random.RandomState(1)
>>> X = rng.randint(5, size=(6, 100))
>>> y = np.array([1, 2, 3, 4, 5, 6])
>>> from sklearn.naive_bayes import MultinomialNB
>>> clf = MultinomialNB(force_alpha=True)
>>> clf.fit(X, y)
MultinomialNB(force_alpha=True)
>>> print(clf.predict(X[2:3]))
[3]
```

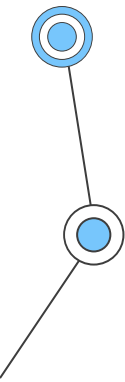


Naive Bayes en Scikit-learn

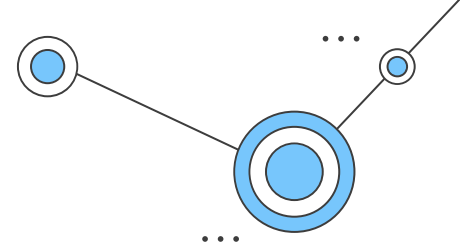


Para características continuas se utiliza `GaussianNB()`, con el que se asume un Likelihood es gaussiano para cada una de ellas

```
>>> from sklearn.datasets import load_iris
>>> from sklearn.model_selection import train_test_split
>>> from sklearn.naive_bayes import GaussianNB
>>> X, y = load_iris(return_X_y=True)
>>> X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=0)
>>> gnb = GaussianNB()
>>> y_pred = gnb.fit(X_train, y_train).predict(X_test)
>>> print("Number of mislabeled points out of a total %d points : %d"
...       % (X_test.shape[0], (y_test != y_pred).sum()))
Number of mislabeled points out of a total 75 points : 4
```



Referencias



- <https://www.freecodecamp.org/espanol/news/como-funcionan-los-clasificadores-naive-bayes-con-ejemplos-de-codigo-de-python/>
- https://scikit-learn.org/stable/modules/naive_bayes.html
- <https://www.youtube.com/watch?v=ugbWqWCcxrg&t=152s>

