Fundamentos y Naive Bayes

Tomás Fontecilla

19 de agosto de 2022

Introducción a Scikit-learn

- Scikit-learn (sklearn) es la librería más utilizada robusta para Machine Learning en Python.
- sklearn provee una selección de herramientas eficientes de machine learning y modelamiento estadístico, incluyendo clasificación, regresión, clustering y reducción dimensional utilizando una interfase consistente de Python.
- Esta librería *mayormente* escrita en Python, utiliza NumPy, SciPy y Matplotlib.

Instalación

Si ya se encunetra instalado NumPy y Scipy, la forma más fácil es:

```
pip install -U scikit-learn
```

Usando conda

```
conda install scikit-learn
```

si no están instalados NumPy y Scipy entonces pueden instalarse con pip o conda.
Otra forma es utilizar **canopy** o **Anaconda**.

Características

Scikit-learn no se preocupa en la manipulación, resumen o carga de datos (para eso está **PANDAS!**) sino en el modelamiento de datos. Algunos de los grupos de modelos más populares son:

- **Algoritmos de aprendizaje supervisado**: SVM, Arboles de decision, Regresión
- **Algoritmos de aprendizaje no supervisado**: analisis factorial, componentes principales
- **clustering**: datos sin catalogar
- **Validación cruzada**
- **Reducción de dimensiones**
- **Métodos de ensamblaje**
- **Extracción de características**
- **Elección de características**

Hiperparámetros

Todo algoritmo, ya sea supervisado o no supervisado, dependerá de parámetros que no tienen relación directa con el algoritmo en si. Son parámetros cuyo valor son usados para controlar el proceso de aprendizaje.

Ejemplos:

- Nivel de tolerancia del error
- Probabilidad de aceptación de una unidad muestral
- Número máximo de iteraciones

Éstos no son parámetros del *algoritmo*.

validación de modelos

la validación de modelos es un método de verificación de qué tan cerca de la realidad están las predicciones. La validación de modelo significa calcular la certeza (u otra medida de evaluación) del modelo que se está entrenando.
Existen varios métodos diferentes que pueden ser usados para validar los modelos de ML.

validación de modelos

Este método es el más comúnmente usado en la validación. Se separa en tres la muestra de trabajo:

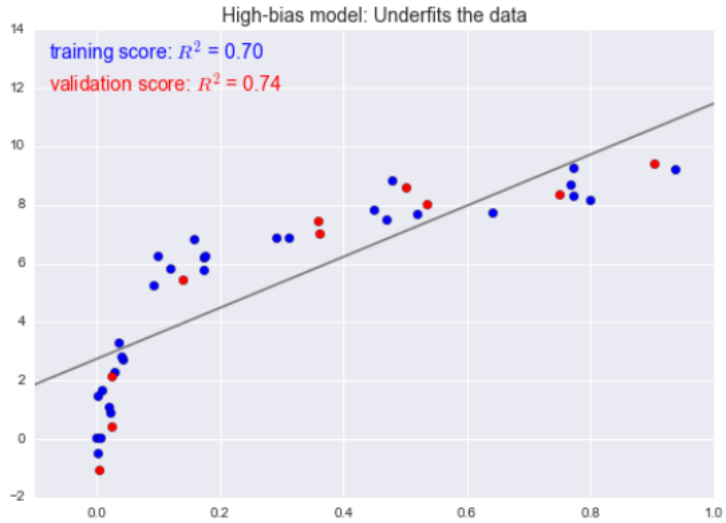
- **Entrenamiento:** Conjunto de datos en el que se entrenará el modelo. Todo el aprendizaje de máquina lo haremos en este dataset.
- **Validación:** Conjunto con el que *afinaremos* el modelo entrenado en el conjunto anterior. Aquí normalmente veremos si cambiamos algún hiperparámetro mejora el modelo *ya seleccionado*.
- **Prueba:** La generalibilidad del modelo se prueba en este conjunto. Es la última etapa de evaluación y muestra si el modelo está listo para ser puesto en producción o no.

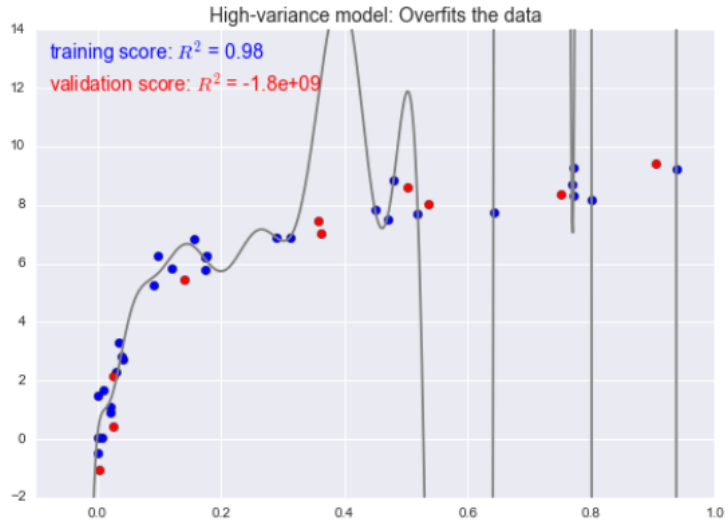
Validación de modelos

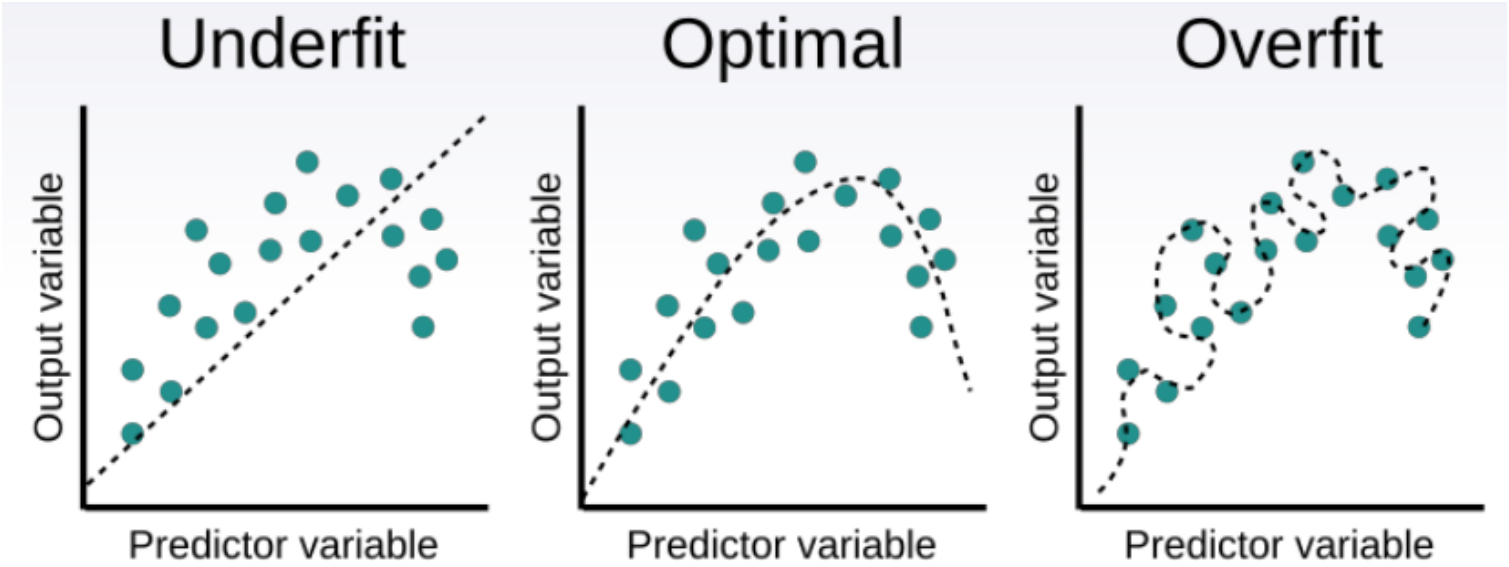
Otras formas de hacer validación:

- Validación Cruzada
- Validación por *k-pliegues*
- Validación por dejar uno fuera

Validación de modelos

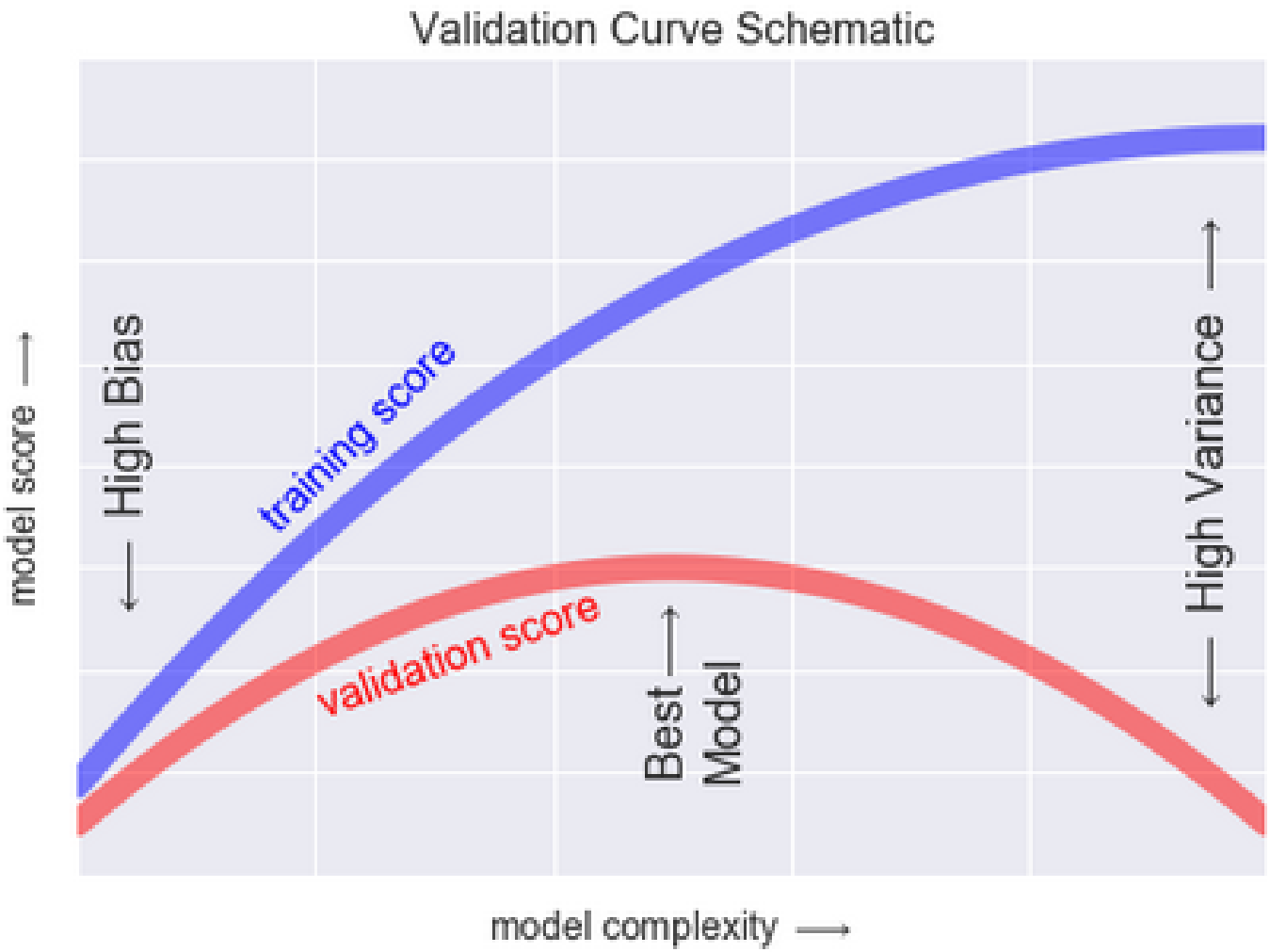






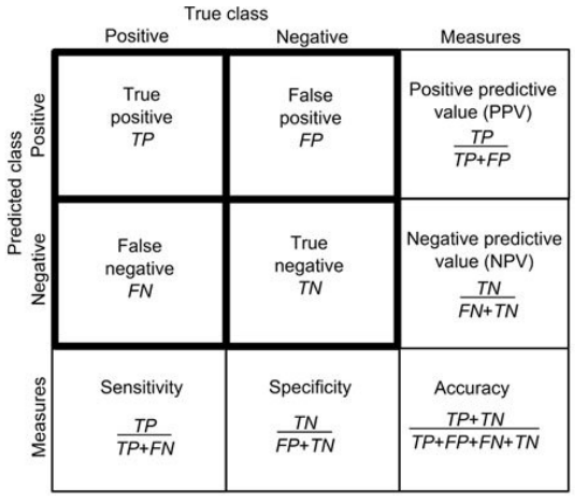
validación de modelos

validación de modelos

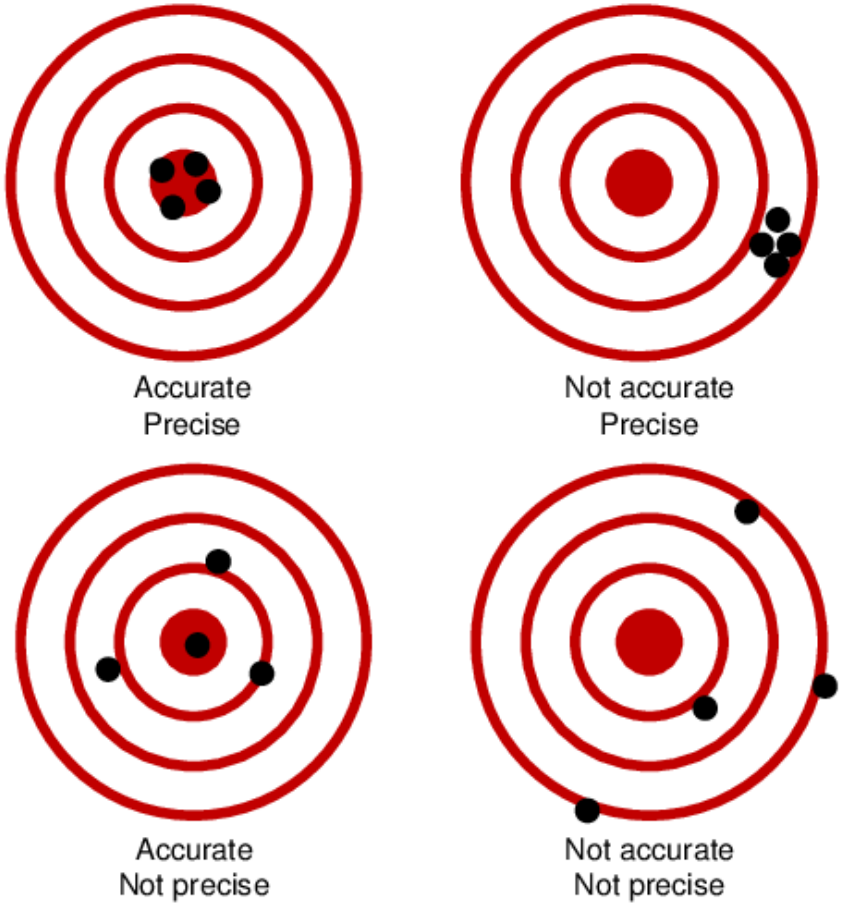


Métricas de Validación

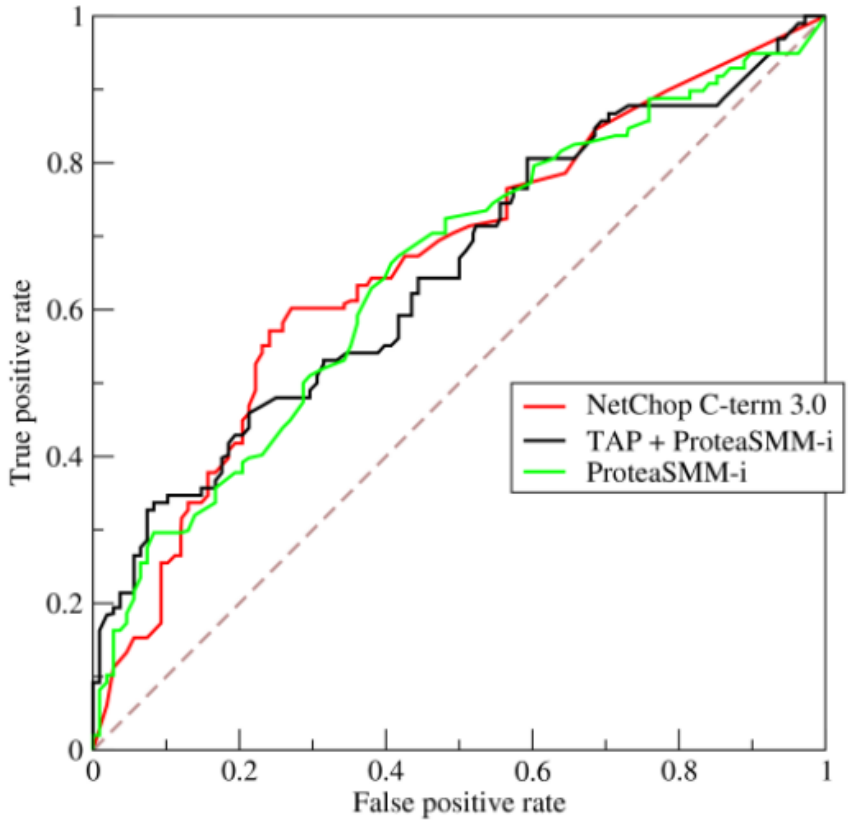
- Precision: La tasa de positivos verdaderos entre los clasificados positivos.
- Recall: La tasa de positivos verdaderos entre los verdaderamente positivos.
- Accuracy: La tasa de clasificación correcta sobre el total.
- F1: la media armónica entre pecisión y recall. Se calcula como
  $F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$



validación de modelos: Diferencia Precisión y Certeza



validación de modelos: Curva ROC



Quote: John Tukey

"La combinación de algunos datos con el deseo imperativo de tener respuesta no asegura tener una respuesta razonable que pueda ser extraída de un cuerpo de datos dado"

Feature Engineering

Buenas características de datos corresponderían a:

- Llevan a comprender los datos
- Retienen información relevante
- Son creadas en base a conocimiento aplicado experto

Algunos errores comunes

- Tratar de automatizar la selección de características
- no poner atención a los caprichos específicos de los datos
- No usar información arbitrariamente

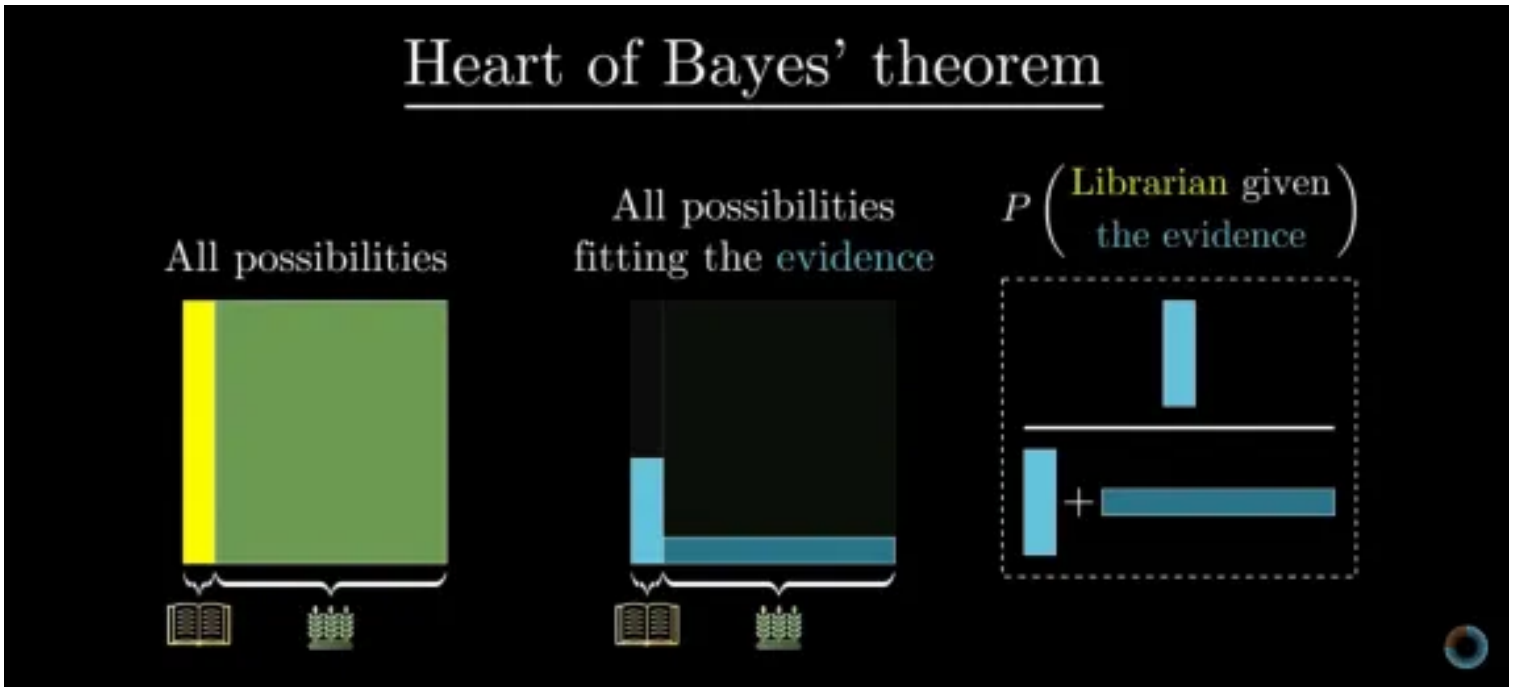
Naive Bayes

El clasificador de Naive Bayes es un modelo de machine learning *probabilista*. Este clasificador, como su nombre lo indica, está basado en el teorema de Bayes:

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

Usando el teorema de Bayes, podemos encontrar la probabilidad que ocurra A dado que B ha ocurrido.
Aquí B es la evidencia y A es la hipótesis. El supuesto que se hace aquí es que *los predictores/características son independientes*. Esto es, la presencia de una característica en particular no afecta la presencia de otra. Por eso se llama "Naive"

Naive Bayes: Corazón del algoritmo



Naive Bayes: Ejemplo

Veamos la problemática de jugar golf. los datos son los siguientes:

	Outlook	Temperature	Humedad	Viento	Jugar Golf
0	Lluvia	Calor	Alta	Falso	No
1	Lluvia	Calor	Alta	Verdadero	No
2	Cubierto	Calor	Alta	Falso	Sí
3	Soleado	Templado	Alta	Falso	Sí
4	Soleado	Frío	Normal	Falso	Sí
5	Soleado	Frío	Normal	Verdadero	No
6	Cubierto	Frío	Normal	Verdadero	Sí
7	Lluvia	Templado	Alta	Falso	No
8	Lluvia	Frío	Normal	Falso	Sí
9	Soleado	Templado	Normal	Falso	Sí
10	Lluvia	Templado	Normal	Verdadero	Sí
11	Cubierto	Templado	Alta	Verdadero	Sí
12	Cubierto	Calor	Normal	Falso	Sí
13	Soleado	Templado	Alta	Verdadero	No

Naive Bayes: Ejemplo - Continuación

- Clasificaremos si el día está bueno para jugar golf, dadas las características del día.
- El primer supuesto que haremos será que cada característica consideraremos que independiente de las otras. Esto es, si la temperatura es Calor, no necesariamente indica que la humedad será alta.
- Otro supuesto es que los predictores tienen un mismo efecto en el resultado. Esto es, un día ventoso no tiene mayor importancia en la decisión de jugar o no al golf.

Según este ejemplo, podemos reescribir el teorema de bayes como:

$$\mathbb{P}(y|X) = \frac{\mathbb{P}(X|y)\mathbb{P}(y)}{\mathbb{P}(X)}$$

donde y es la variable (jugar golf) y X son las características.

Naive Bayes: Ejemplo - resolución

Así, como $X = (x_1, x_2, x_3, ..., x_n)$ donde i = 1,,...n son las características, podemos sustituir X en la formula expandiendo por *regla de la cadena*:

$$\mathbb{P}(y|x_1, ..., x_n) = \frac{\mathbb{P}(x_1|y)\mathbb{P}(x_2|y)...\mathbb{P}(x_n|y)\mathbb{P}(y)}{\mathbb{P}(x_1)\mathbb{P}(x_2)...\mathbb{P}(x_n)}$$

Como el denominador no cambia, podemos decir que permanece estático y por proporcionalidad se remueve.

$$\mathbb{P}(y|x_1, ..., x_n) \propto \mathbb{P}(y)\Pi_{i=1}^{n}\mathbb{P}(x_i|y)$$

Por último, clasificamos el valor de y donde se maximice la probabilidad, esto es:

$$y = argmax_y \mathbb{P}(y)\Pi_{i=1}^{n}\mathbb{P}(x_i|y)$$

Naive Bayes: Ejemplo - Resultados

Hay que calcular una serie de probabilidades de eventos *dado* la condición,

Característica-evento(X)	Golf - No	Golf - Si	$\mathbb{P}(X\|Y = No)$	$\mathbb{P}(X\|Y = Si)$
Clima = Soleado	2	3	0.4	0.33
Clima = Lluvia	3	2	0.6	0.22
Clima = Cubierto	0	4	0	0.45
Temperatura = Calor	2	2	0.4	0.22
Temperatura = Templado	2	4	0.4	0.45
Temperatura = Fío	1	3	0.2	0.33
Humedad = Alta	4	3	0.8	0.33
Humedad = Normal	1	6	0.2	0.67
Viento = False	2	6	0.4	0.67
Viento = True	3	3	0.6	0.33

Así, la probabilidad del evento en i = 10, correspondiente a Lluvia, Templado,Normal,Verdadero está dado por una de dos posibilidades: $y = No$ o $y = Si$

$$\mathbb{P}(y = No|X) \propto 0{,}36 \cdot 0{,}6 \cdot 0{,}4 \cdot 0{,}2 \cdot 0{,}6 = 0{,}0103$$
$$\mathbb{P}(y = Si|X) \propto 0{,}64 \cdot 0{,}22 \cdot 0{,}45 \cdot 0{,}67 \cdot 0{,}33 = 0{,}014$$

El máximo es 0.014 luego clasificamos que sí es posible jugar golf. (Una nota importante, que al normalizar, esto corresponde a un 57,46 %)

Naive bayes: Ejemplo - Validación

Para validar, veremos la matriz de confusión:

		Clase Real		
		Positivo	Negativo	Medida
Clase Predicha	Positivo	6	1	0.8571
	Negativo	3	4	0.5714
	Medida	0.667	0.8	0.714

$F_1 = 0{,}75$

Tipos de Clasificador Naive Bayes

- **Naive Bayes Multinomial:** Este es el más utilizado para clasificación de documentos, si pertenece a una categoría. Usualmente los predicotres son frecuencia de palabras presentes en el documento.
- **Naive Bayes Bernoulli:** Similar al anterior, pero los predictores son booleanos. La clase a predecir es usualmente "Si/No"
- **Naive Bayes Gaussiano:** Cuando los predictores toman valores continuos y no son discretos, se puede asumir que los valores son muestreados de una distribución gaussiana

Conclusión

Naive bayes es un algoritmo muy utilizado en análisis de sentimiento, filtrado de spam, sistemas recomendadores, etc. Usualmente es rápido y fácil de implementar pero su mayor desventaja es que requiere que los predictores sean independientes, cuyo caso en vida real no es así, perjudicando el desempeño del algoritmo.