

Machine Learning

Módulo 4 - parte 1

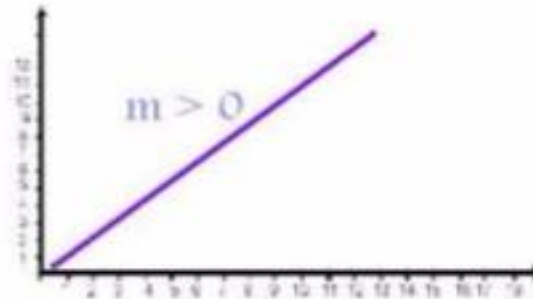
Actualizado el 23/10/25

Regresión lineal

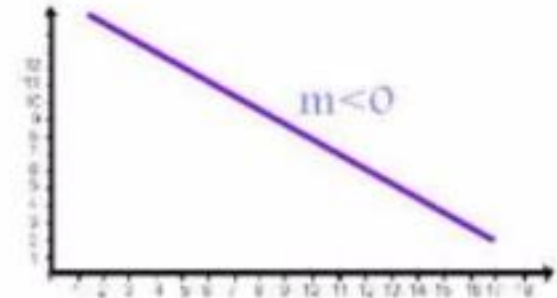
Pendiente de una recta

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

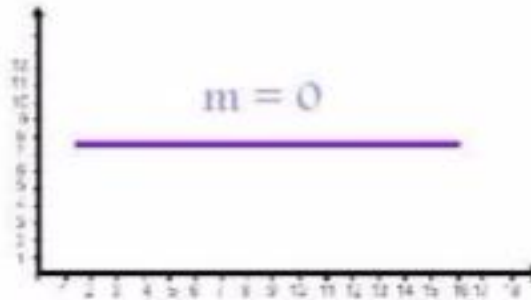
Pendiente = positiva



Pendiente negativa

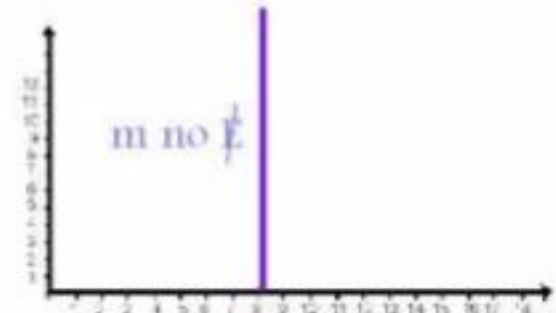


$m = 0$



Pendiente = 0

m no \exists

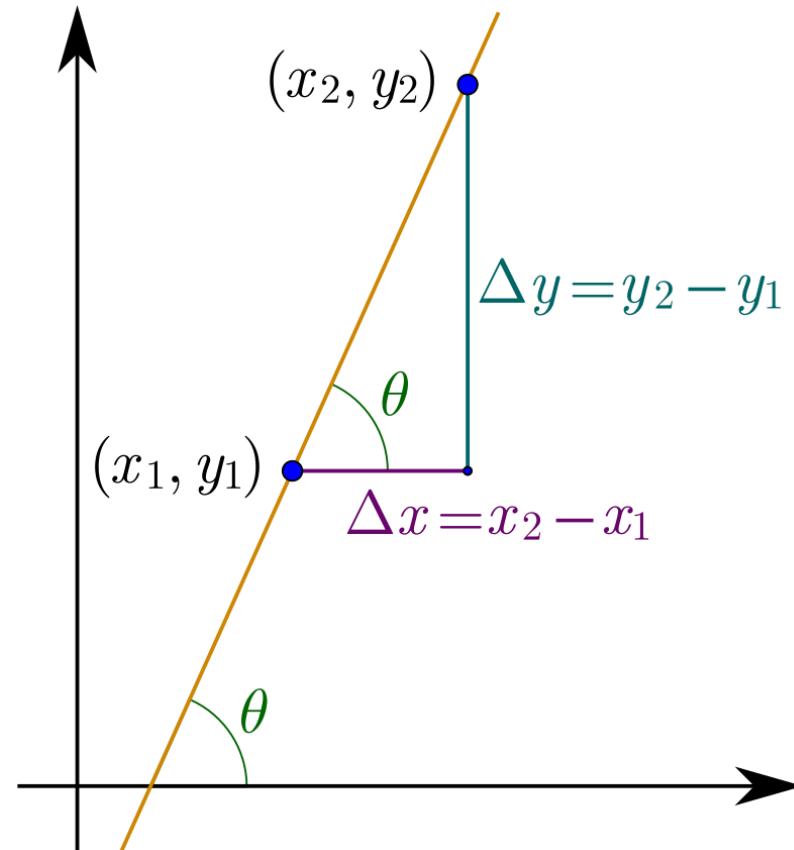


Pendiente indefinida

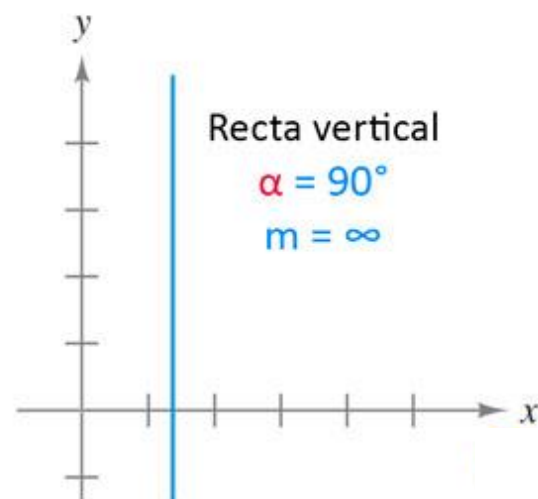
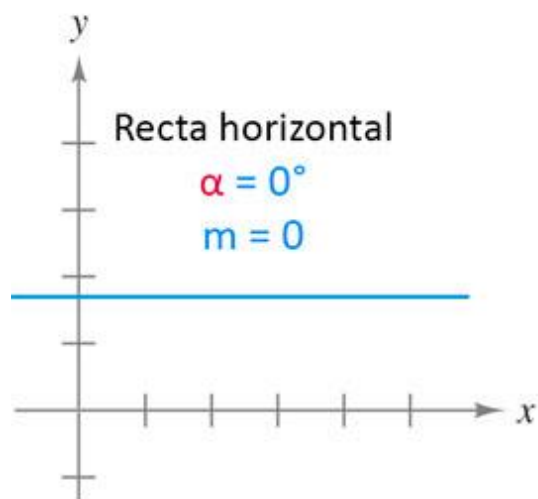
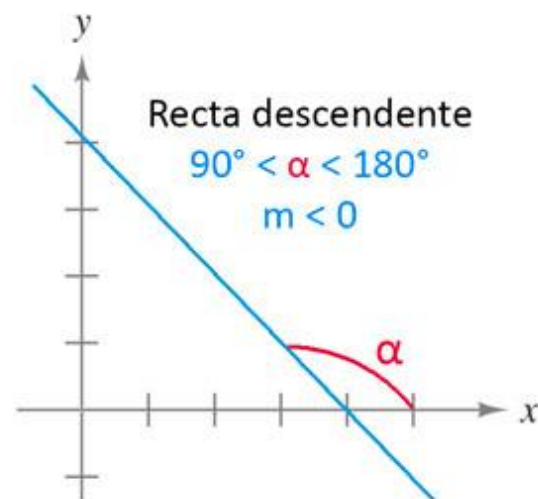
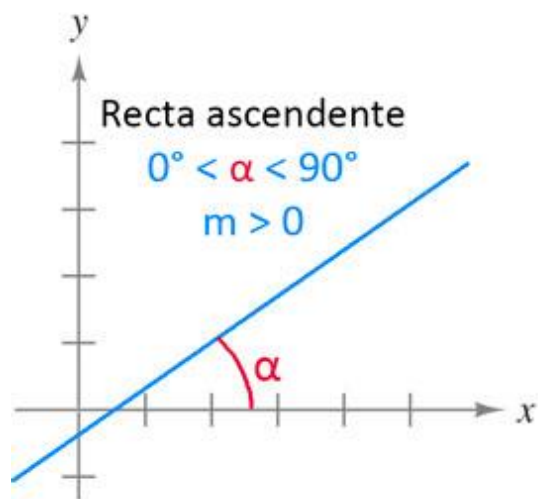
Pendiente de una recta

Es una medida de su **inclinación** y su **dirección** con respecto al eje horizontal (x)

$$m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$$



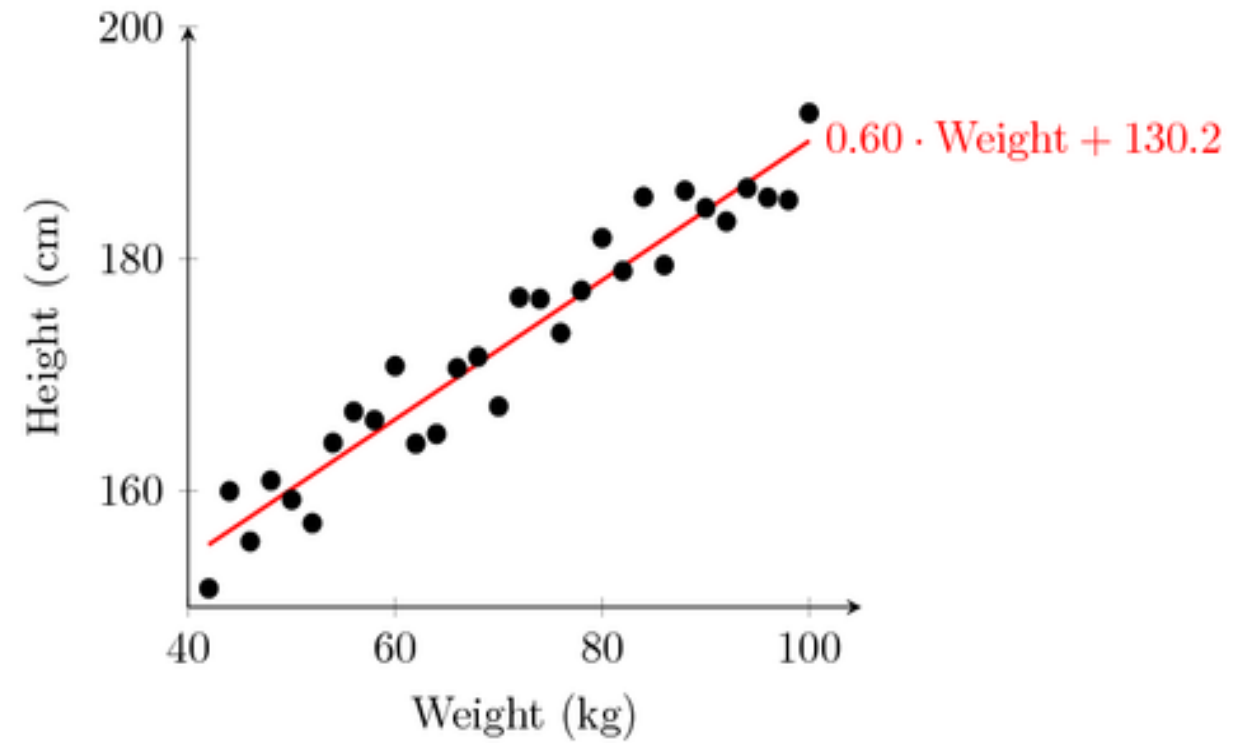
Pendiente de una recta



Regresión Lineal

- Crea una **función lineal** dado un grupo de observaciones
- Usa esta función para **predecir** valores no observados
- El tiempo no tiene relación en este tipo de modelo

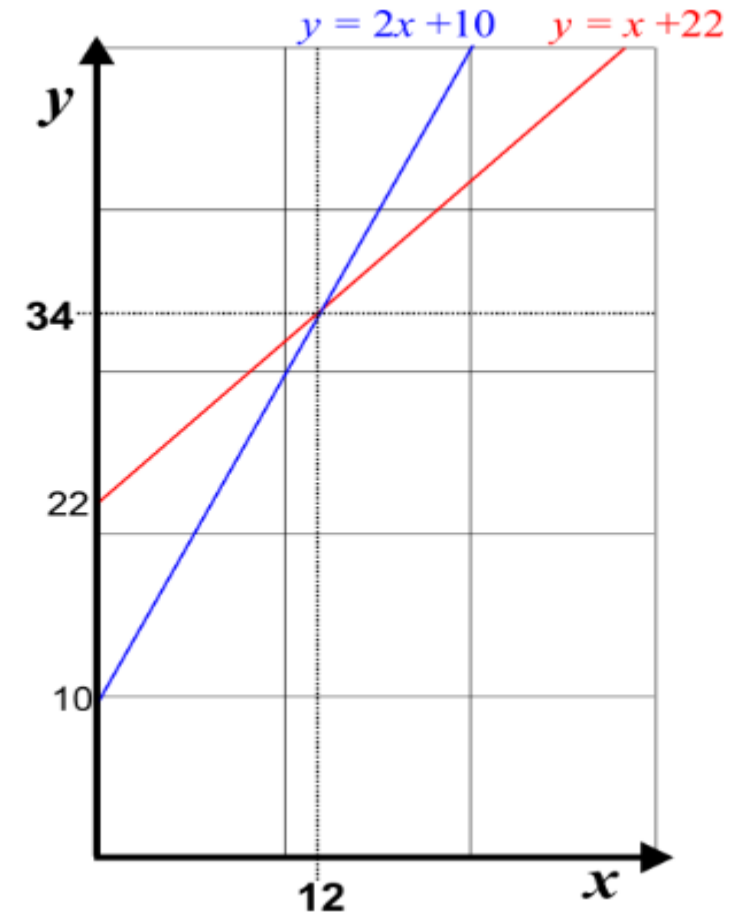
Regresión Lineal



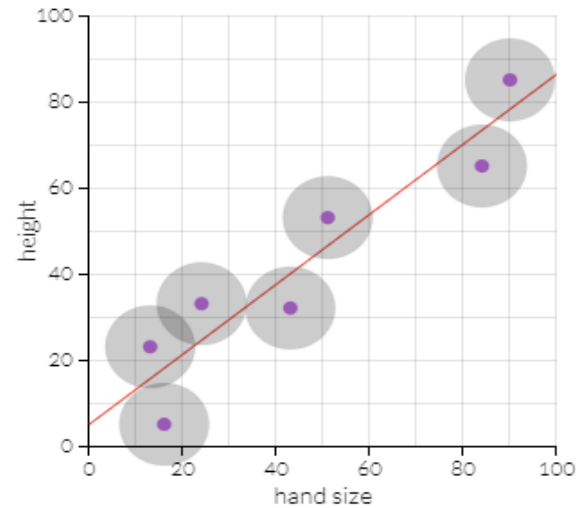
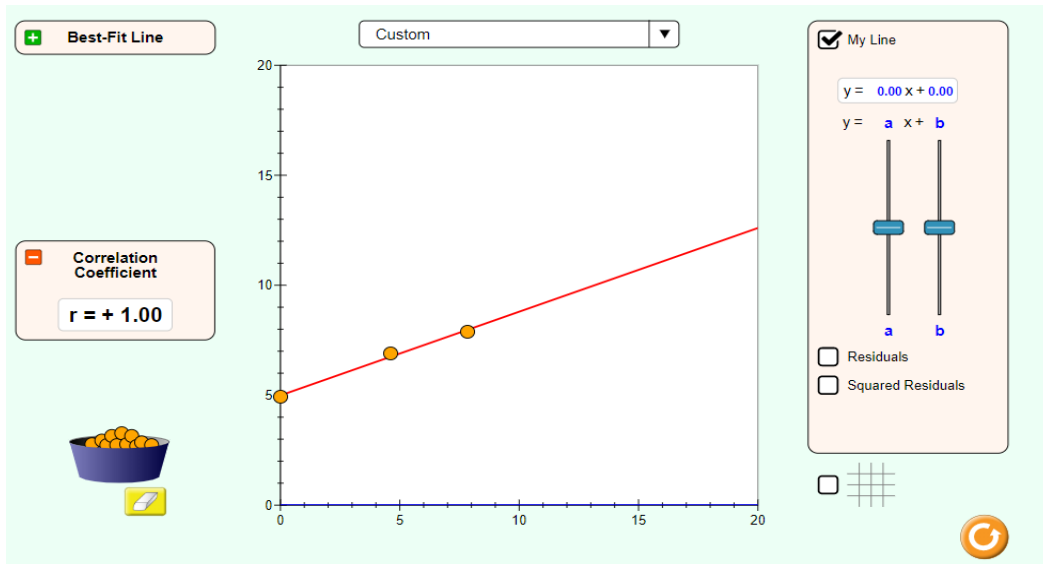
Regresión Lineal

Ecuación de la recta:

$$y = mx + b$$



Regresión Lineal

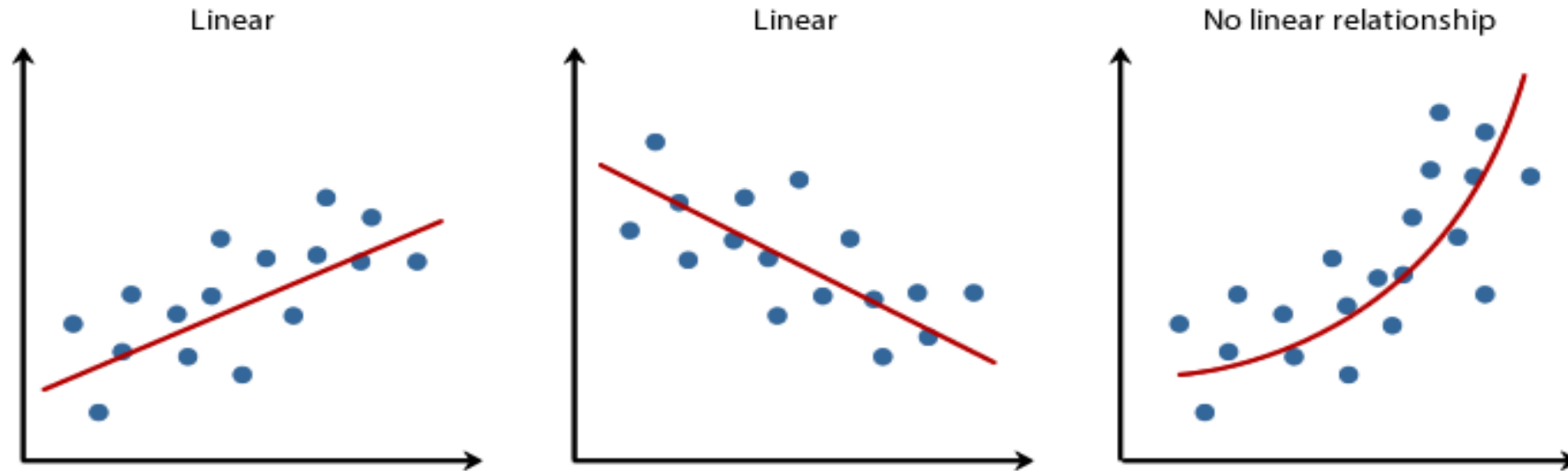


Beta 1 - The y-intercept of the regression line.

$$5.00 + 0.81 * \text{hand size} = \text{height}$$

Beta 2 - The slope of the regression line.

Regresión Lineal



Regresión Lineal

Cálculo:

- Mínimos Cuadrados Ordinarios o Ordinary Least Squares (OLS)
- Minimiza los errores (al cuadrado) entre cada punto y la línea
- $y = mx + b$

Regresión Lineal

Pendiente: $m = r \frac{S_y}{S_x}$

- S_x : Desviación estándar en X
- S_y : Desviación estándar en Y
- r : Coeficiente de Correlación

$$r = \frac{1}{N-1} \sum \left(\frac{x_1 - \bar{x}}{S_x} \right) \left(\frac{y_1 - \bar{y}}{S_y} \right)$$

Intersección: $b = \bar{y} - m\bar{x}$

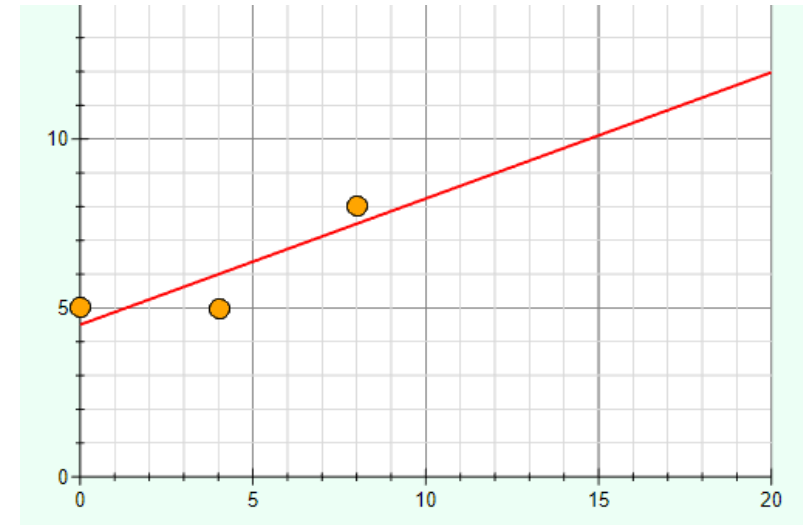
Regresión Lineal

$$m = 0,867 \frac{1,73}{4} = 0.37$$

$$b = 6 - (4 * 0.37) = 6 - 1.48 = 4.52$$

$$y = 0.37 * x + 4.52$$

Abrir: regresión-lineal



Regresión Lineal

Ejercicio: Al ejercicio de altura y tamaño manos calcular la regresión lineal con Excel y Python. Graficar los resultados.

R^2 o Coeficiente de determinación

- Sirve para testear mi modelo
- R^2 mide:

La fracción de la variación total en Y capturada por el modelo. Dice qué tan bien el modelo se ajusta a los datos reales

R^2 o Coeficiente de determinación

Mide el **porcentaje de la variabilidad** de una variable que es explicada por la otra. Por ejemplo, un R^2 de 0.64, lo que significa que el 64% de la variabilidad en la variable dependiente es explicada por el modelo de regresión.

R^2

Interpretación:

- Se mide en un rango de 0 a 1
- 0 es peor (nada de la varianza es capturada por el modelo), 1 es mejor (toda la varianza es capturada)

$$r^2 = \frac{RSS}{SSTO} = \frac{\sum(\hat{y} - \bar{y})^2}{\sum(y - \bar{y})^2}$$

$$R^2$$

- RSS: “Regression Sum of Squares” es la “suma de los cuadrados de la regresión” y cuantifica la distancia entre la línea de regresión y la media en y .
- SSTO: “Total Sum of Squares” es la “suma total de los cuadrados” y cuantifica la distancia entre los valores reales y la media en y .

Relación entre R^2 y r

En una regresión lineal simple, el R-cuadrado es simplemente el **coeficiente de correlación al cuadrado**:

$$R^2 = r^2$$

- El **coeficiente de correlación (r)** va de -1 a 1. El signo indica la **dirección** de la relación (positiva o negativa), y la magnitud indica la **fuerza** de la relación.
- El **R^2** va de 0 a 1. Al ser un valor al cuadrado, siempre es positivo o cero. Esto significa que R-cuadrado no te dice la dirección de la relación, solo su **fuerza**.

$$R^2$$

- Abrir: regresión-lineal.ipynb
- Ejercicio: Calcular R^2 para el ejercicio anterior con Excel y Python.

Regresión multivariable

Regresión multivariable

- Más de una variable influye en la que estoy interesado
- Por ejemplo: predecir el precio de un coche basado en sus atributos (marca, modelo, cantidad de puertas, etc.)

Regresión multivariable

$$\text{precio} = \alpha + \beta_1 km + \beta_2 edad + \beta_3 puertas$$

- Los coeficientes indican cuán importante es cada factor (si todos los datos están normalizados)
- Se pueden quitar los que no importan
- Se puede usar también r cuadrado
- Los factores se tratan de manera independiente cada uno. Si son dependientes entre ellos, esto no se refleja en el modelo (kilometraje/edad).

Pandas DataFrame

Column Label/ Header		0	1	2	3	4	
Index Label		Name	Age	Marks	Grade	Hobby	Column Index
0	S1	Joe	20	85.10	A	Swimming	
1	S2	Nat	21	77.80	B	Reading	
2	S3	Harry	19	91.54	A	Music	
3	S4	Sam	20	88.78	A	Painting	Row
4	S5	Monica	22	60.55	B	Dancing	
				Column		Element/ Value/ Entry	

Pandas DataFrame - resumen

Pandas DataFrame

- **pd.DataFrame():** Crear un DataFrame desde cero (diccionarios, listas, etc.).
- **pd.read_csv(), pd.read_excel(), pd.read_json(), pd.read_sql():** Importar datos desde archivos o bases de datos.
- **df.head():** Muestra las primeras n filas (por defecto 5).
- **df.tail():** Muestra las últimas n filas (por defecto 5).
- **df.shape:** Devuelve una tupla con el número de filas y columnas.
- **df.describe():** Genera estadísticas descriptivas (conteo, media, desviación estándar, min, max, cuartiles) de las columnas numéricas.
- **df['NombreColumna']** o **df.NombreColumna:** Seleccionar una sola columna (devuelve una Serie).
- **df[['ColA', 'ColB']]:** Seleccionar múltiples columnas (devuelve un DataFrame).

Pandas DataFrame

- **df.loc[]**: Selección por etiquetas (nombres de índice y columna).
Ejemplo: `df.loc[0, 'ColA']` o `df.loc[3:5, ['ColA', 'ColB']]`
- **df.iloc[]**: Selección por posición entera (índices de fila y columna).
Ejemplo: `df.iloc[0, 0]` o `df.iloc[3:6, 0:2]`
- **df[df['ColA'] > 10]**: Filtrar filas donde la columna 'ColA' cumple una condición.
- **.sum(), .mean(), .median(), .min(), .max(), .std()**: Calculan la suma, media, mediana, mínimo, máximo, y desviación estándar
- **.groupby('Columna')**: Agrupa el DataFrame utilizando valores de una o más columnas y permite realizar operaciones de agregación en los grupos resultantes (ej. `df.groupby('Pais')['Poblacion'].sum()`).

Pandas DataFrame

- **df.concat([df1, df2]):** Concatena DataFrames (apilando filas o columnas).
- **df.merge(df1, df2, on='clave'):** Fusiona DataFrames (similar a las operaciones JOIN en SQL) en base a una o más columnas clave.

Regresión multivariable

- OLS: Ordinary Least Squares
- Abrir: `regresión-multivariable.ipynb`
- Ejercicios:
 - Para valores de un coche nuevo, predecir su precio y comprobar en Excel que sea correcto.

Regresión multivariable

- Ejercicio: dados unos datos (advertising.csv) sobre inversión en publicidad (TV, radio y diario) y las ventas correspondientes buscar:
 - Dividir los datos en 80%/20%, train/test
 - La regresión lineal para publicidad en TV y ventas
 - Graficar los resultados
 - Calcular R cuadrado
 - Añadir el resto de las variables para practicar regresión multivariable
 - Calcular R cuadrado y cuál de las variables es la menos importante.

Normalización y Estandarización

¿Por qué necesitamos transformar los datos?

- Los **modelos estadísticos** y de **machine learning** son sensibles a la **escala de las variables**.
- Sin estas técnicas:
 - Una variable con **números grandes** puede *dominar* a las demás.
 - **Comparar** métricas diferentes se vuelve *injusto o poco claro*.

Ejemplo:

Si una empresa analiza **ingresos mensuales** (en miles) y **nivel de satisfacción** (escala 1-5), los ingresos dominarán el análisis.

Normalización (min-max scaling)

Escala los datos **entre 0 y 1**.

Fórmula:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

Ejemplo:

Ventas de empleados: 200, 400 y 800 unids.

Valor mínimo = 200

Valor máximo = 800.

Para 400:

$$(400-200)/(800-200) = 200/600 \approx 0.33$$

Para 800:

$$(800-200)/(800-200) = 600/600 = 1$$

Interpretación: Ahora todos los datos están en una escala entre 0 y 1, lo que facilita comparaciones.

Estandarización (z-score)

Convierte los datos en una escala con **media 0** y **desviación estándar 1**.

Fórmula:

$$Z = \frac{X - \mu}{\sigma}$$

Ejemplo:

Edades: 20, 30, 40.

Media = 30

Desviación estándar ≈ 10 .

Para 40:

$$Z = (40-30)/10 = 1$$

→ está 1 desviación estándar por encima de la media.

Interpretación: Nos dice cuán lejos está un valor respecto al promedio en términos relativos.

Sistemas de recomendaciones

¿Qué son los sistemas de recomendaciones?

amazon.es
Prueba Prime

Todos los departamentos

Enviar a Jorge
Barcelona 08037

Todos los departamentos

Volver a comprar

Amazon.es de Jorge

Ofertas

Chollos

Cheques regalo

Vender

Hola Jorge

Cuenta y listas

Pedidos

Suscríbete a Prime

Cesta

Mi Amazon.es

Mi historial de navegación


Recomendaciones para ti

Mejorar mis recomendaciones


Mi perfil público

Más información


Recomendado para ti, Jorge




Bebés
13 ARTÍCULOS



Deportes
27 ARTÍCULOS

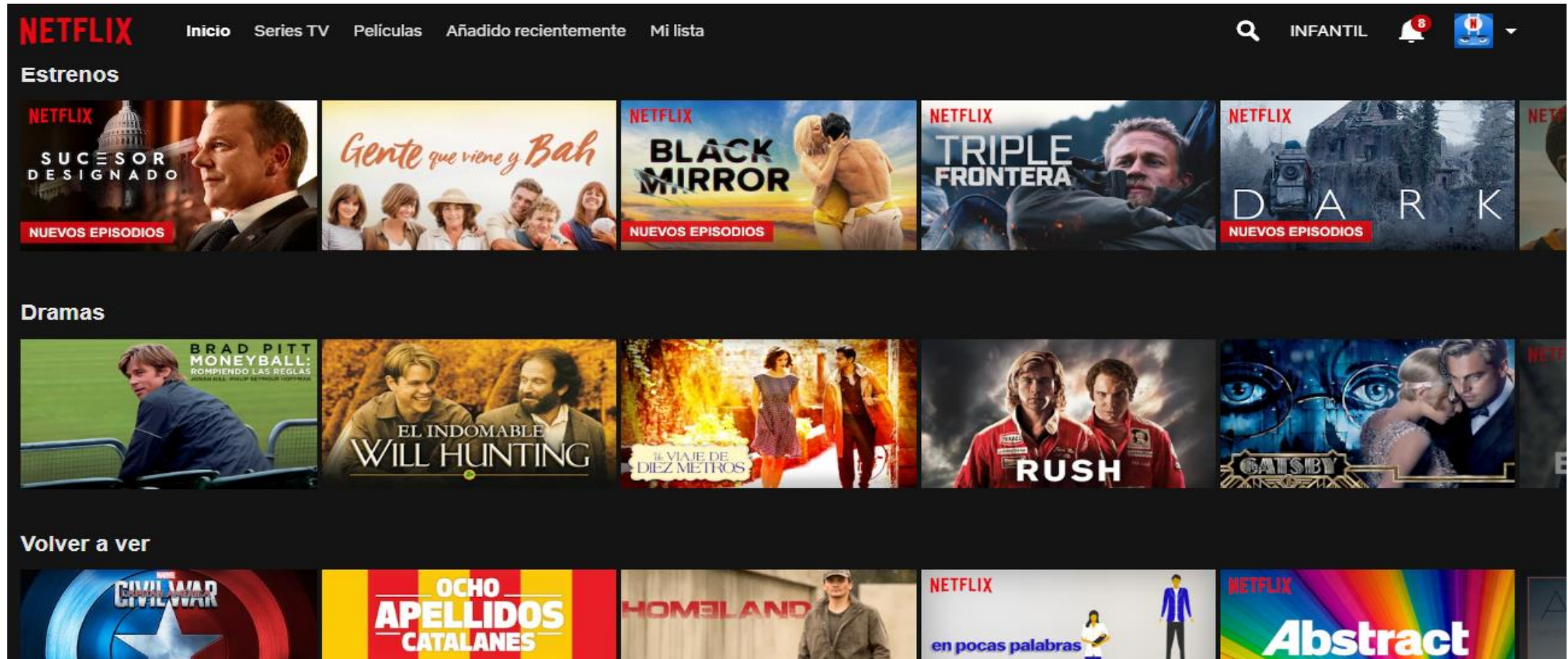


Hogar y cocina
25 ARTÍCULOS



Libros
100 ARTÍCULOS

¿Qué son los sistemas de recomendaciones?



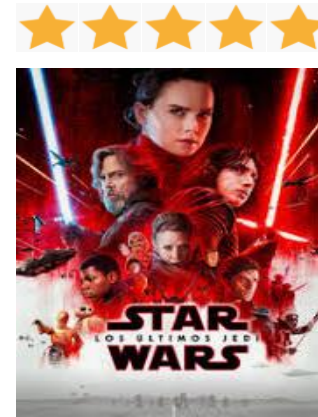
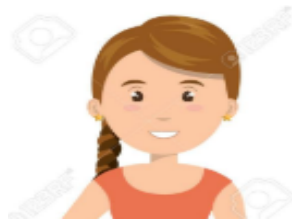
Sistemas de recomendaciones

- Filtrado colaborativo basado en usuarios (User-Based Collaborative Filtering)
- Filtrado colaborativo basado en items (Item-Based Collaborative Filtering)

Filtrado colaborativo basado en usuarios

- Crear una matriz de cosas que cada usuario a comprado/visto/calificado.
- Calcular puntajes de similaridad entre usuarios
- Encontrar usuarios similares a uno.
- Recomendar cosas que ellos han comprado/visto/calificado pero que yo no.

Filtrado colaborativo basado en usuarios



Filtrado colaborativo basado en usuarios



Filtrado colaborativo basado en usuarios

Problemas:

- Las personas cambian
- Generalmente hay más personas que cosas (problema de cálculo)
- Las personas pueden engañar al sistema para tratar de promocionar un producto (cuentas falsas)

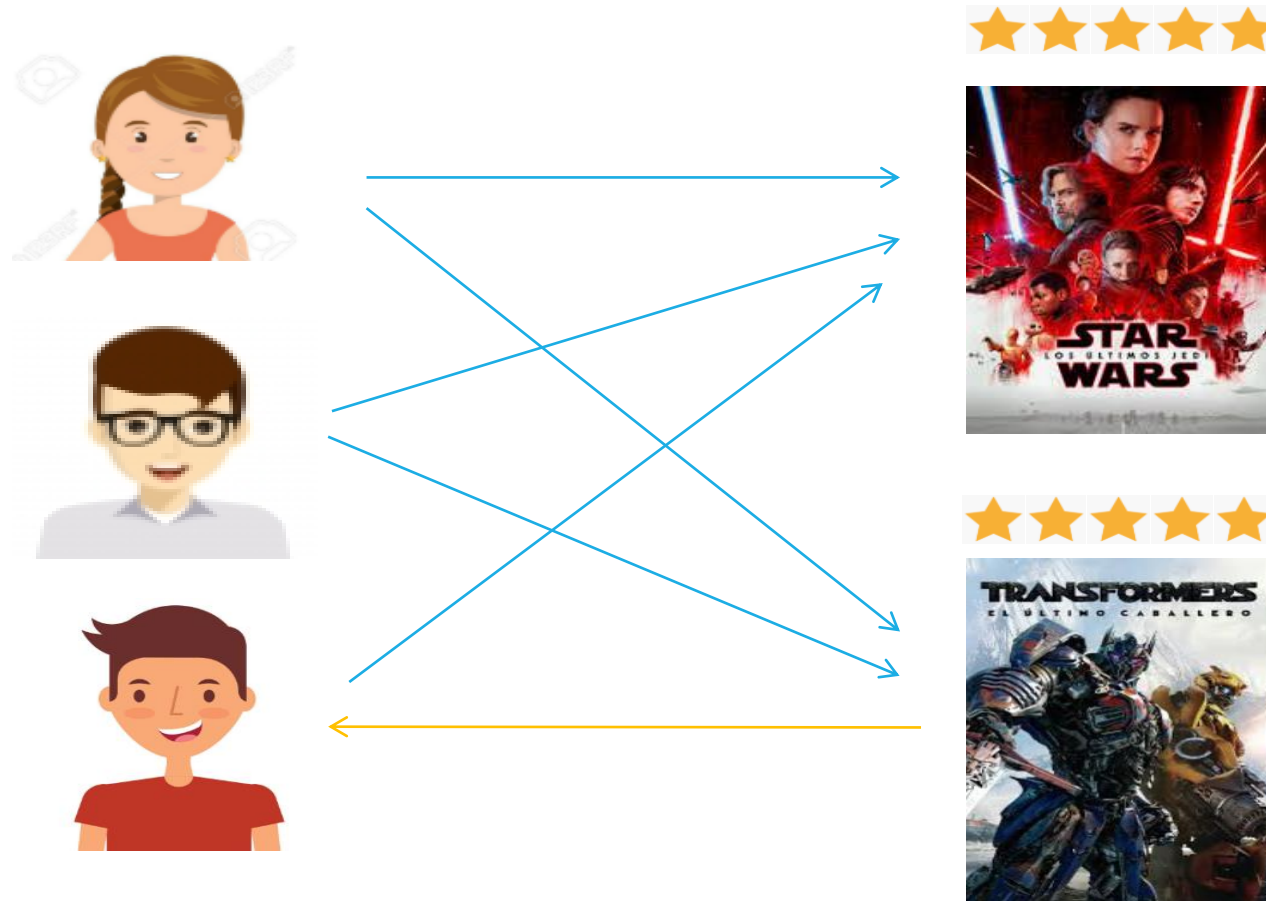
Filtrado colaborativo basado en items

- Un item (por ejemplo una película) siempre será igual, no cambia
- Hay menos items que personas
- Es más difícil de engañar el sistema

Filtrado colaborativo basado en items

- Encontrar cada par de películas que han sido vistas por la misma persona
- Medir la similaridad (la correlación) de sus calificaciones en todos los usuarios que vieron a las dos

Filtrado colaborativo basado en items



Filtrado colaborativo basado en items

- Vamos a crear el código en Python para un sistema de recomendaciones basado en cosas que de “películas similares” usando los datos reales de MovieLens de IMDB.
- Visitar <https://grouplens.org/>
- Abrir: `pelis-similares.ipynb` y `recomendaciones-pelis.ipynb`

Ejercicio

- Vamos a hacer un SdR basado en items para recomendar animé.
Descargar los datos de Kaggle:
 - <https://www.kaggle.com/CooperUnion/anime-recommendations-database>
- Primero encontrar similitudes entre los animé.
- Finalmente generar recomendaciones a partir de los animé que ha visto una persona.
- El usuario de pruebas debe haber visto:
 - Hunter X hunter 2011 (ID 11061) con 10 estrellas
 - School dates (ID 2476) con 1 estrella

Ejercicio

- Crear una API para el ejercicio de las sugerencias de animé que tenga la siguiente funcionalidad:
 - Obtener recomendaciones
 - Entrenar el algoritmo
 - Obtener la versión
 - Testear el algoritmo
- Crear algún consumidor de esa API. Puede ser una aplicación de consola, una web, una app, etc.

Ejercicio

Pasos para crear la funcionalidad de obtener recomendaciones:

- Crear una ruta de tipo GET con la URL /obtener-recomendaciones que tenga como parámetros los animés y las calificaciones del usuario. Deberá retornar en formato JSON los animés recomendados.
- Crear una aplicación de consola que permita ingresar los animés y sus calificaciones. Además hará la llamada a la API para obtener las recomendaciones.

Probabilidad

Probabilidad

- La probabilidad es una medida de la **certidumbre** asociada a un suceso o evento futuro y suele expresarse como un número entre 0 y 1 (o entre 0 % y 100 %).
- Un suceso puede ser **improbable** (con probabilidad cercana a cero), **probable** (probabilidad intermedia) o **seguro** (con probabilidad uno).

Ejemplos

Lanzamiento de una moneda.

- ¿Cuál es la probabilidad de que salga cara?

- $$P(A) = \frac{\text{Número de casos favorables}}{\text{Número de casos posibles}}$$

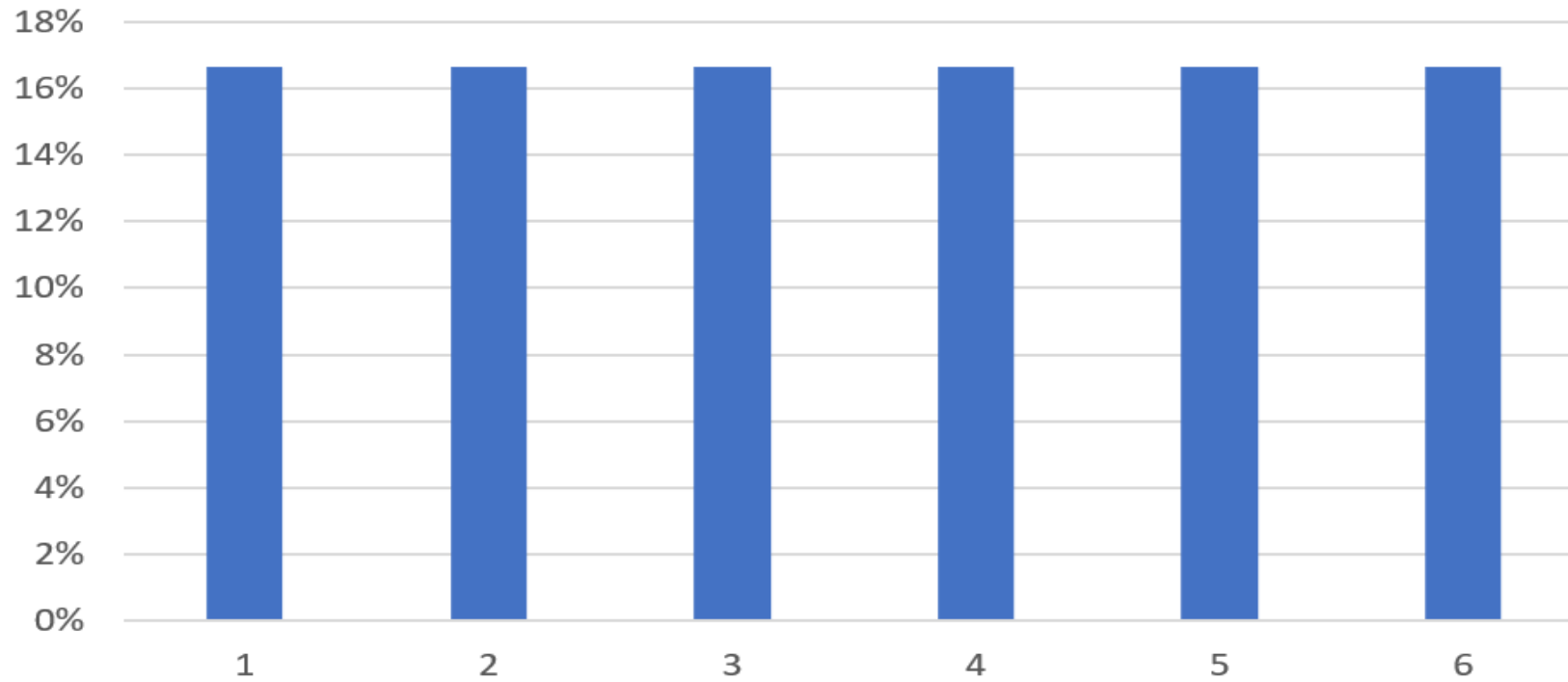
- $$P(\text{"salga cara"}) = \frac{1}{2} = 0,5 = 50\%$$

Ejemplos

Lanzamiento de un dado.

- ¿Cuál es la probabilidad de que salga cinco?
- $P(\text{"salga cinco"}) = \frac{1}{6} = 0.16 = 16.66\%$
- ¿Cuál es la probabilidad de que salga par?

Ejemplos



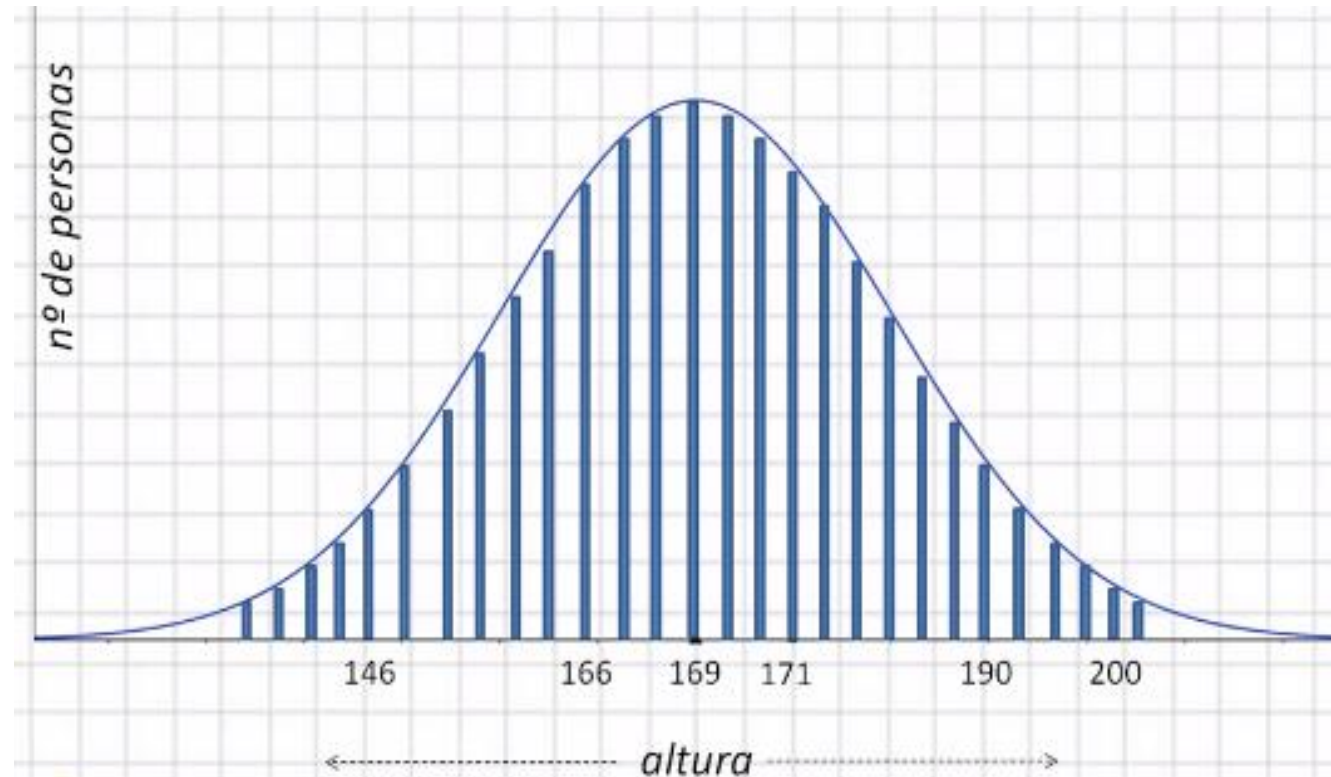
Distribución de probabilidad para un dado

Funciones de probabilidad

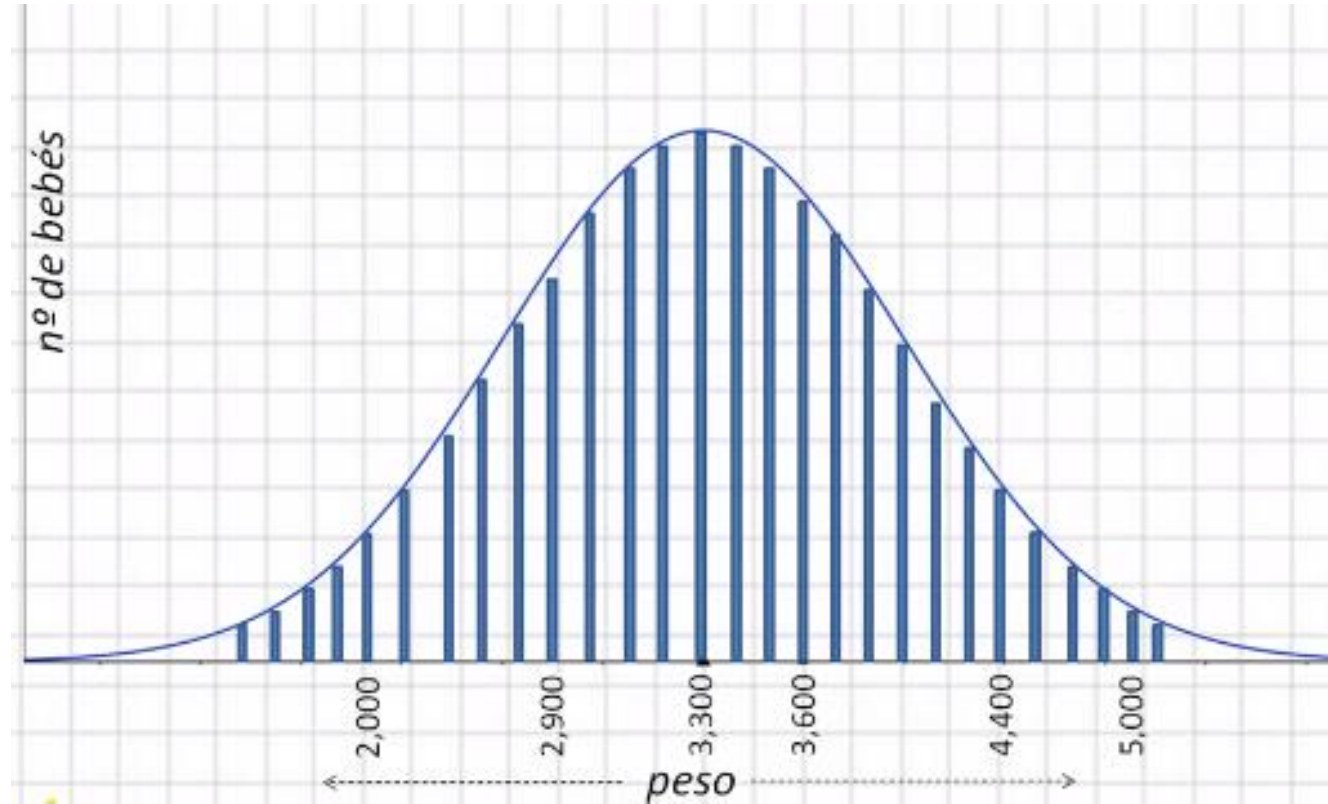
Distribución normal

- Ejemplo: histograma de alturas
- Para datos continuos
- Da la probabilidad para un dato de caer dentro del rango de un valor de probabilidad dado

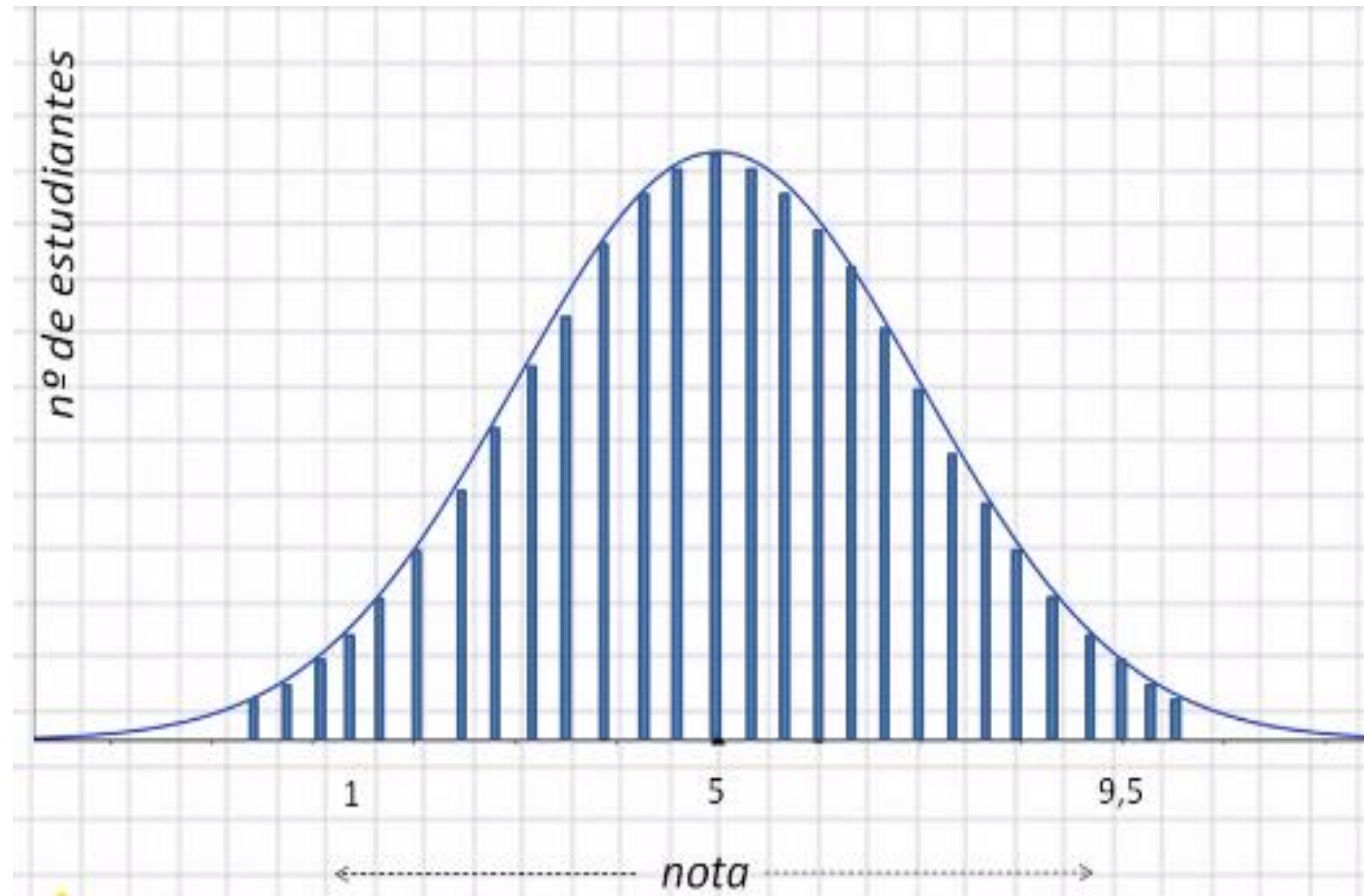
Distribución normal



Distribución normal

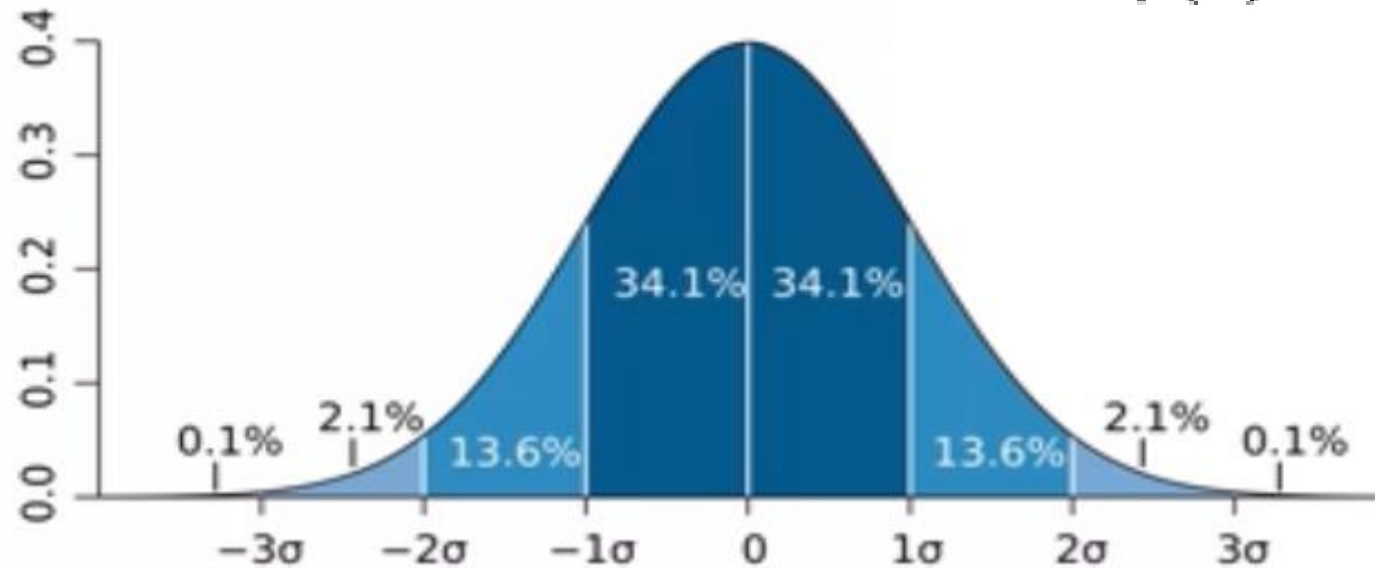


Distribución normal



Distribución normal

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$$



Por ejemplo: existe un 34,1% de probabilidad de que un valor caiga dentro del rango de la media y la media + una desviación estándar.

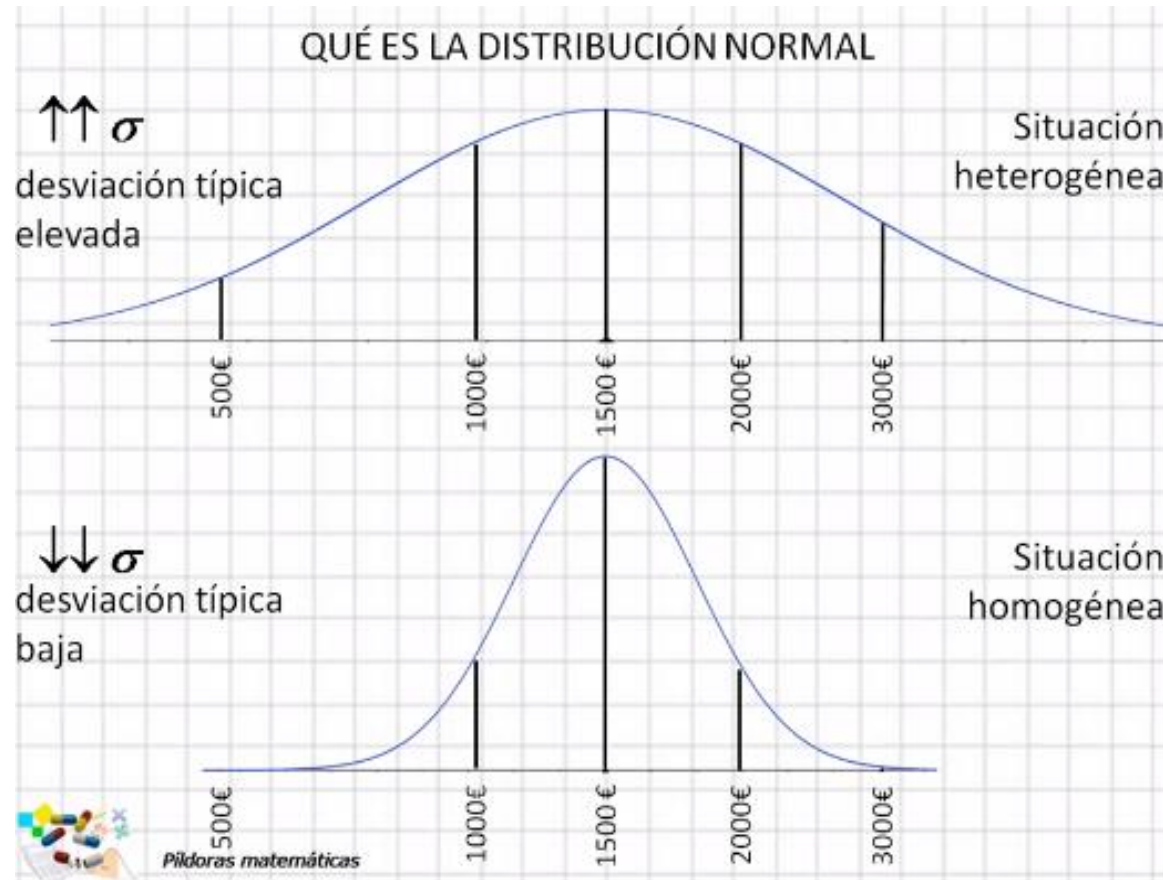
Distribución normal

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Fórmula completa

- μ (mu) es la **media** de la distribución, que define el centro de la campana.
- σ (sigma) es la **desviación estándar**
- e es la base del logaritmo natural

Distribución normal



Otras distribuciones

Abrir: distribuciones.ipynb

- Uniforme: lanzamiento de un dado
- Exponencial: probabilidad de correcto funcionamiento en dispositivos

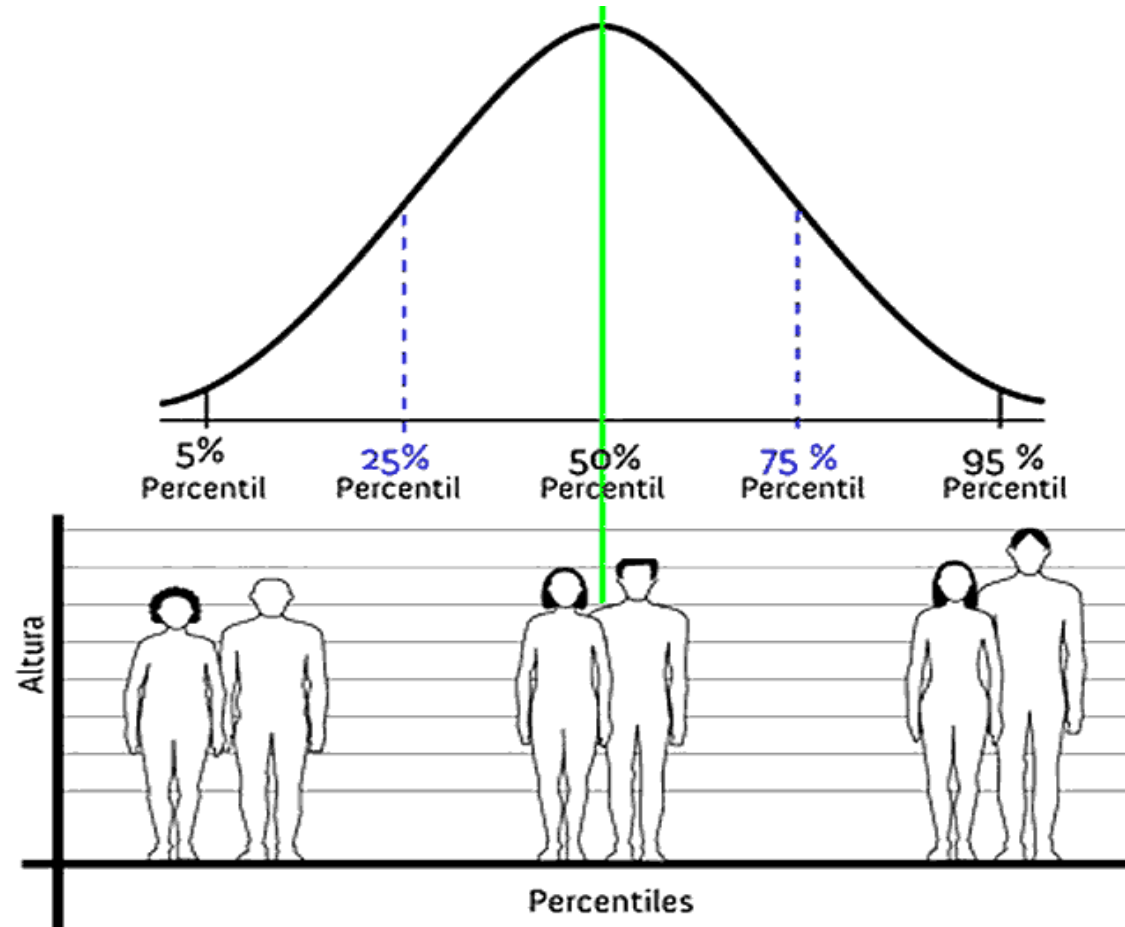
Ejercicio: dar al menos un ejemplo real para cada distribución.

Percentiles y momentos

Percentil

Valor que divide un conjunto ordenado de datos estadísticos de forma que un porcentaje de tales datos sea inferior a dicho valor.

Percentil



Percentil

- Abrir: percentiles.ipynb
- Ejercicio: dado el valor determinado, obtener su percentil.

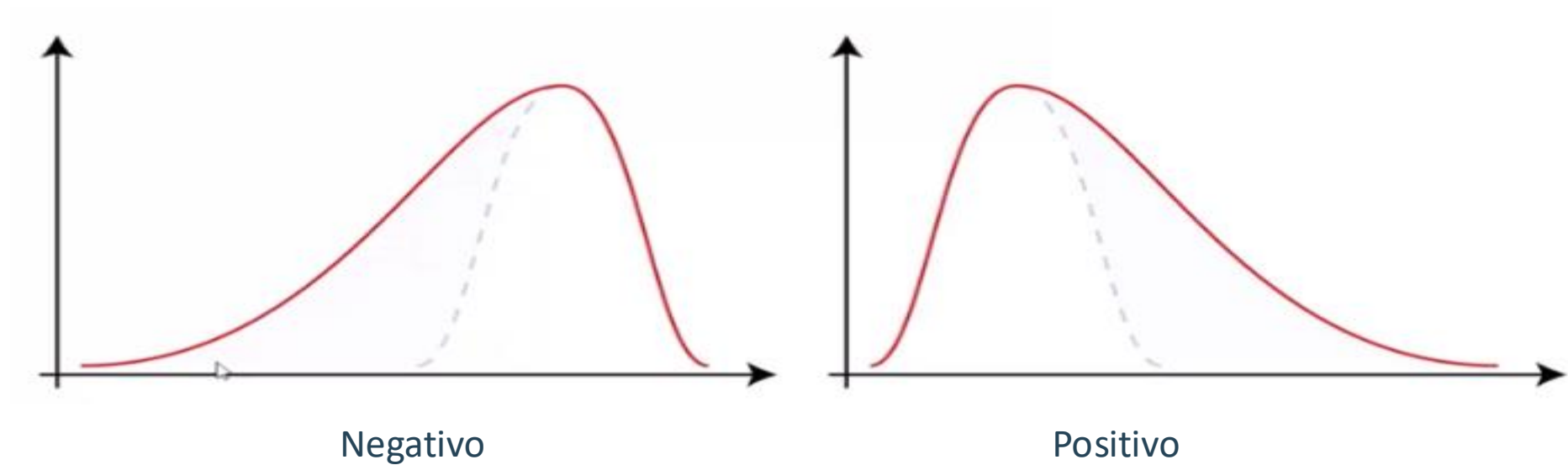
Momentos

Medidas cuantitativas de la forma de una función de probabilidad

- El 1er momento es la media
- El 2do momento es la varianza

Momentos

- El 3er momento denota la asimetría y es llamado “sesgo”

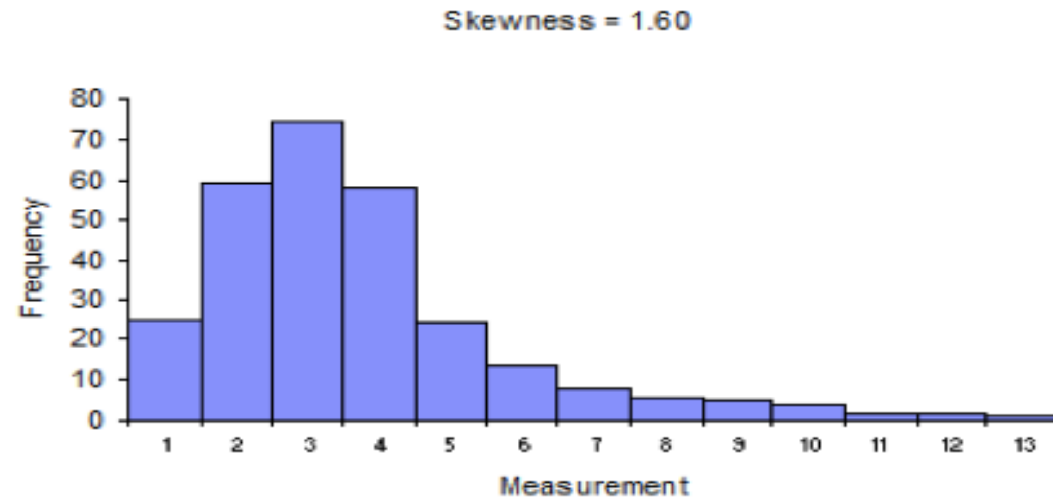
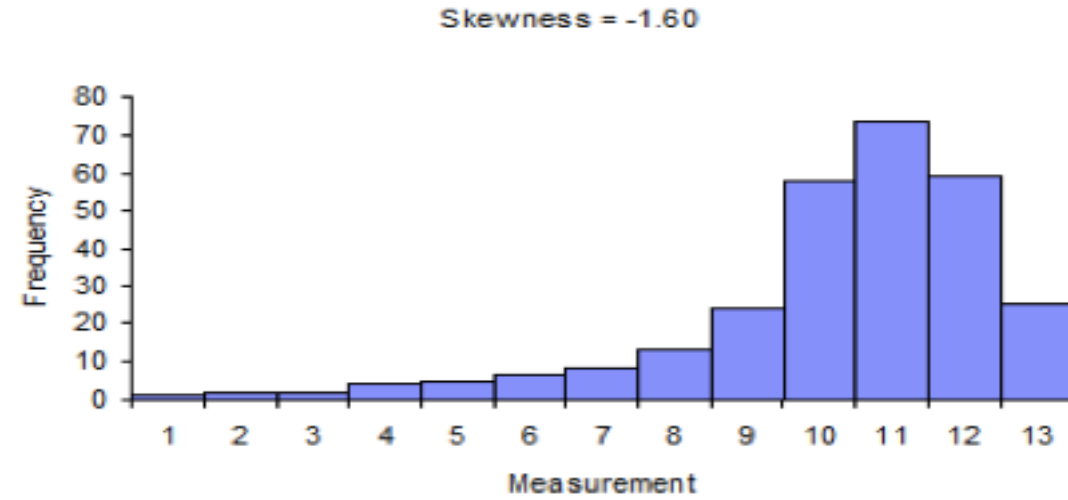


Momentos

El sesgo:

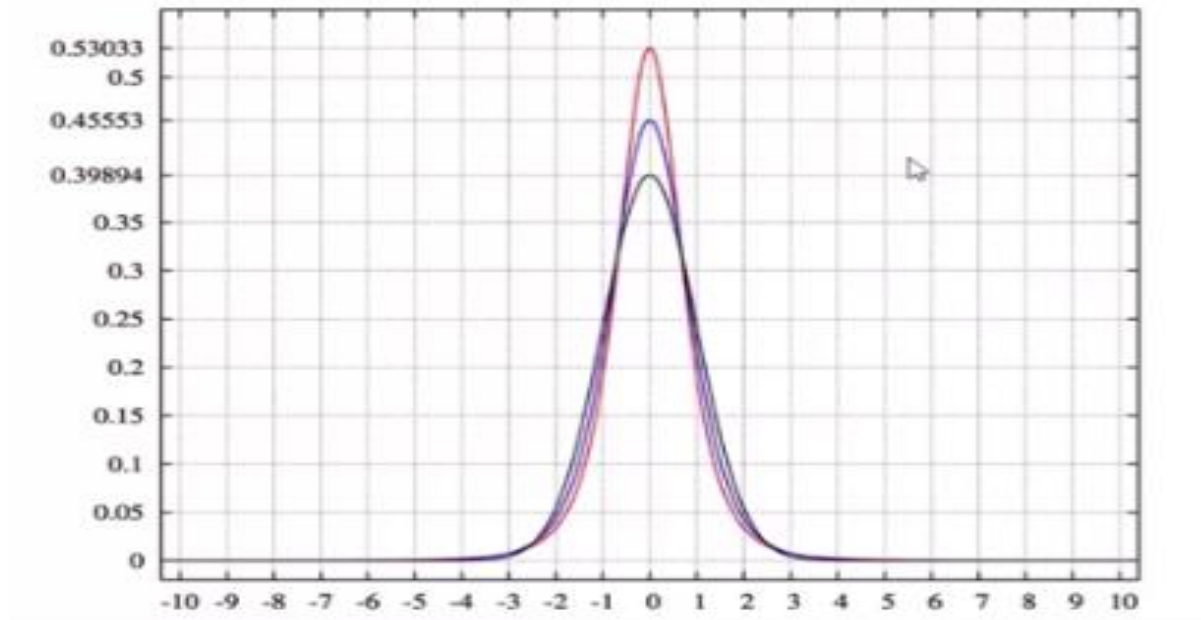
- Puede ser positivo o negativo
- Si es 0 los datos son perfectamente simétricos
 - Si es menor a -1 o mayor a 1: altamente sesgado
 - Si está entre -1 y -0,5 o entre 0,5 y 1: moderadamente sesgado
 - Si está entre -0,5 y 0,5: aproximadamente simétrica

Momentos

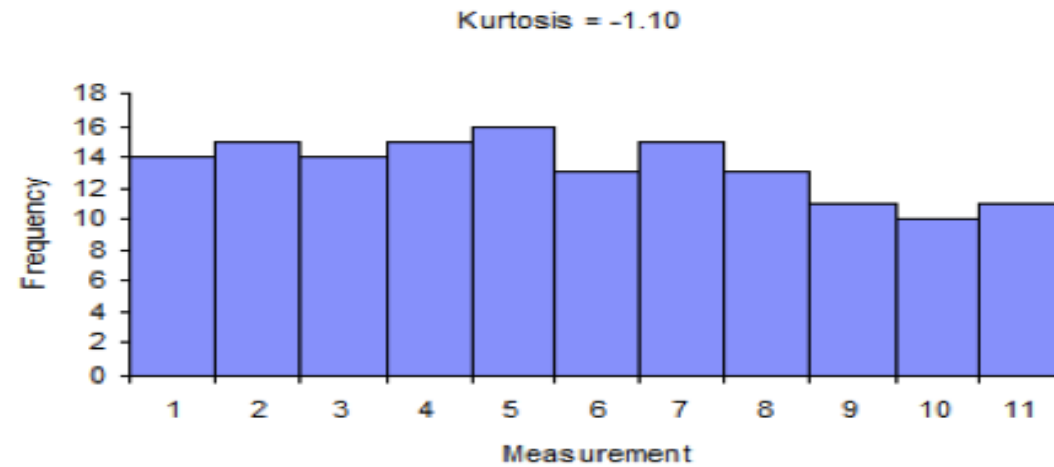
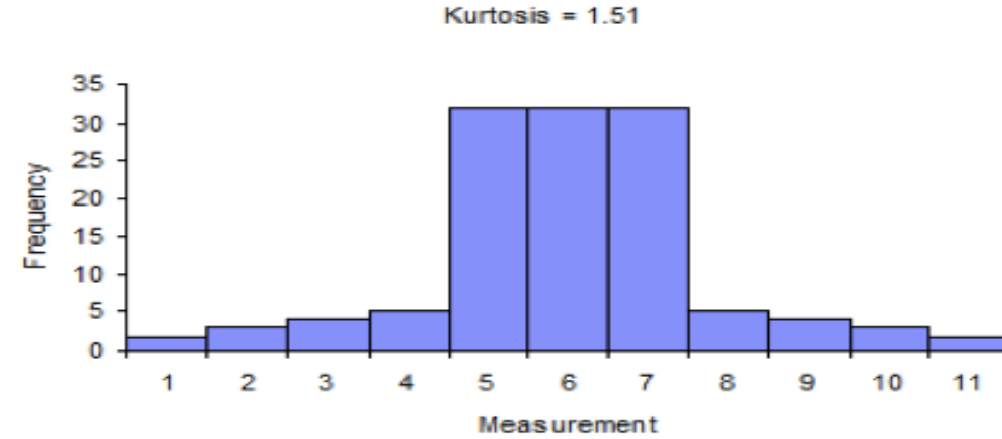


Momentos

- El 4to momento se llama “curtosis”
 - Cuan puntiaguda es la forma, comparada con una distribución normal
 - Picos más altos tienen mayor curtosis



Momentos



Momentos

Abrir: `momentos.ipynb`

Probabilidad condicional

Probabilidad condicional

- La probabilidad condicional mide la probabilidad de un determinado suceso conociendo información previa sobre otro suceso.
- Ejemplo:
 - Se sabe que el 50% de la población fuma y que el 10% fuma y es hipertensa. ¿Cuál es la probabilidad de que un fumador sea hipertenso?

Probabilidad condicional

- Si tenemos dos eventos que **dependen** entre si ¿cuál es la probabilidad de que ocurrido un evento suceda el otro?
- $P(A,B)$ es la probabilidad de que los dos ocurran de manera independiente.
- $P(B|A)$ la probabilidad de que ocurra B dado que A ha ocurrido (implica una dependencia)

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

Probabilidad condicional

- Se sabe que el 50% de la población fuma y que el 10% fuma y es hipertensa. ¿Cuál es la probabilidad de que un fumador sea hipertenso?
- A = fumador, B = hipertenso

$$P(B|A) = \frac{P(A, B)}{P(A)} = \frac{0.1}{0.5} = 0.2 = 20\%$$

Probabilidad condicional

- Le doy a mis estudiantes dos exámenes. 60% aprueban los dos, pero el primero es más fácil, así que el 80% lo aprueba. ¿Qué porcentaje de estudiantes que pasó el primero también pasó el segundo?
- A = aprobar el primer examen
- B = aprobar el segundo examen
- $P(B|A)$ es la probabilidad de B dado A

$$P(B|A) = \frac{P(A, B)}{P(A)} = \frac{0.6}{0.8} = 0.75 = 75\%$$

Probabilidad condicional

- **IMPORTANTE:** No tiene por qué haber una **relación causal** o **temporal** entre A y B . A puede preceder en el tiempo a B , sucederlo o pueden ocurrir simultáneamente. A puede causar B , viceversa o no.
- Las relaciones causales o temporales son nociones que no pertenecen al ámbito de la probabilidad. Pueden desempeñar un papel o no, dependiendo de la interpretación que se le dé a los eventos.

El Teorema de Bayes

El Teorema de Bayes

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

La probabilidad de A dado B, es la probabilidad de A multiplicada por la probabilidad de B dado A, sobre la probabilidad de B.

El Teorema de Bayes: diagnóstico médico

Imagina que existe una enfermedad rara que afecta a 1 de cada 10.000 personas. Existe una prueba médica para detectar esta enfermedad, pero no es perfecta:

- **Sensibilidad:** La prueba detecta correctamente la enfermedad en el 99% de los casos cuando la persona realmente la tiene. Probabilidad de que la prueba sea positiva si la persona está enferma:

$$P(\textit{Positivo}|\textit{Enfermo}) = 0.99$$

- **Especificidad:** La prueba da un resultado negativo correctamente en el 95% de los casos cuando la persona está sana. Probabilidad de que la prueba sea negativa si la persona está sana:

$$P(\textit{Negativo}|\textit{Sano}) = 0.95$$

- Esto significa que la probabilidad de un **falso positivo** (la prueba es positiva, pero la persona está sana) es del 100% menos el 95%:

$$P(\textit{Positivo}|\textit{Sano}) = 1 - P(\textit{Negativo}|\textit{Sano}) = 1 - 0.95 = 0.05$$

El Teorema de Bayes: ejemplo

- **Pregunta:** Si una persona da positivo en la prueba, ¿cuál es la probabilidad de que realmente tenga la enfermedad?
- Aquí es donde el Teorema de Bayes es muy útil. Queremos calcular la **probabilidad a posteriori** de que la persona esté enferma, dado que la prueba dio positivo:

$$P(Enfermo|Positivo)$$

El Teorema de Bayes: ejemplo

Datos:

- $P(Enfermo)$: Probabilidad previa de tener la enfermedad (la prevalencia) = $1/10000 = 0.0001$
- $P(Sano)$: Probabilidad previa de estar sano = $1 - P(Enfermo) = 1 - 0.0001 = 0.9999$
- $P(Positivo|Enfermo)$: Probabilidad de que la prueba sea positiva si se está enfermo (sensibilidad) = 0.99
- $P(Positivo|Sano)$: Probabilidad de que la prueba sea positiva si se está sano (tasa de falsos positivos) = 0.05

Ley de la probabilidad total

Calcular la probabilidad total positiva se refiere a determinar la probabilidad de que un **evento ocurra**, considerando **todas las posibles formas** en que ese evento puede manifestarse. En esencia, estamos **sumando** las probabilidades de cada uno de esos caminos o escenarios que conducen al evento de interés.

$$P(A) = P(A|B_1)P(B_1) + P(A|B_2)P(B_2) + \dots + P(A|B_n)P(B_n)$$

Donde:

- $P(A)$ es la probabilidad total de que ocurra el evento **A**.
- $P(A|B_i)$ es la probabilidad condicional de que ocurra **A** dado que ocurrió B_i .
- $P(B_i)$ es la probabilidad de que ocurra el evento B_i .

El Teorema de Bayes: ejemplo

$$P(\text{Enfermo} \mid \text{Positivo}) = \frac{P(\text{Positivo} \mid \text{Enfermo}) \times P(\text{Enfermo})}{P(\text{Positivo})}$$

Para calcular $P(\text{Positivo})$, la probabilidad total de que la prueba sea positiva, usamos la ley de probabilidad total:

$$P(\text{Positivo}) = P(\text{Positivo} \mid \text{Enfermo}) \times P(\text{Enfermo}) + P(\text{Positivo} \mid \text{Sano}) \times P(\text{Sano})$$

$$P(\text{Positivo}) = (0.99 \times 0.0001) + (0.05 \times 0.9999)$$

$$P(\text{Positivo}) = 0.000099 + 0.049995 \approx 0.050094$$

El Teorema de Bayes: ejemplo

Ahora aplicamos el Teorema de Bayes:

$$P(\text{Enfermo} \mid \text{Positivo}) = \frac{0.99 \times 0.0001}{0.050094}$$

$$P(\text{Enfermo} \mid \text{Positivo}) = \frac{0.000099}{0.050094} \approx 0.001976$$

El Teorema de Bayes: conclusión

- A pesar de que la persona dio **positivo** en la prueba, la probabilidad de que realmente tenga la enfermedad es solo de aproximadamente el **0.1976%**.
- Esto puede parecer contraintuitivo al principio, pero se debe a que la **enfermedad es muy rara**. La gran mayoría de los positivos provienen de personas sanas (falsos positivos), porque hay muchas más personas sanas que enfermas en la población. El Teorema de Bayes nos ayuda a ajustar nuestra creencia inicial (la baja probabilidad previa de tener la enfermedad) basándonos en la nueva evidencia (el resultado positivo de la prueba).

Clasificadores bayesianos
ingenuos

Clasificadores bayesianos ingenuos

Los métodos bayesianos ingenuos son un conjunto de algoritmos de aprendizaje supervisado basados en la aplicación del teorema de Bayes con la suposición "**ingenua**" de la independencia condicional entre cada par de características, dado el valor de la variable de clase.

Tipos de clasificadores bayesianos ingenuos

- **Naive Bayes Gaussiano:** es el más adecuado para datos continuos. Asume que las características de cada clase siguen una **distribución normal (o gaussiana)**. Por ejemplo, en un problema para clasificar el sexo de una persona, si se usan variables como la altura o el peso, este clasificador sería apropiado, ya que estas características tienden a distribuirse de forma normal.
- **Naive Bayes Multinomial:** se utiliza para datos que representan **conteo de frecuencia**, como los que se encuentran en la clasificación de texto. Asume que las características siguen una **distribución multinomial**, donde la ocurrencia de cada característica (por ejemplo, una palabra) es un resultado de un ensayo. Es el clasificador de referencia para el análisis de sentimientos y la clasificación de documentos.
- **Naive Bayes de Bernoulli:** es apropiado para datos **binarios o booleanos**. Funciona asumiendo que las características son variables de tipo **Bernoulli**, es decir, que solo pueden tomar dos valores (0 o 1). Se utiliza, por ejemplo, para clasificar documentos donde la presencia o ausencia de una palabra es lo que importa, en lugar de su frecuencia.

Clasificador bayesiano ingenuo multinomial

- En el contexto de Multinomial Naive Bayes, la palabra "**multinomial**" se refiere a una **distribución multinomial**, que es una generalización de la distribución binomial.
- El término "multinomial" significa que hay **más de dos resultados posibles** para un solo evento. En el caso de Multinomial Naive Bayes, el "evento" es la aparición de una palabra dentro de un documento. En lugar de tener solo dos resultados (como en el lanzamiento de una moneda, cara o cruz), hay muchos resultados posibles, uno para cada palabra en el vocabulario.
- Por ejemplo, al clasificar un documento de texto, cada palabra que aparece en el documento es una "prueba". El "resultado" de esta prueba no es solo "palabra presente" o "palabra ausente", sino más bien "la palabra es 'el'", "la palabra es 'es'", o "la palabra es 'gato'", con una probabilidad para cada posible palabra. La **distribución multinomial** modela el recuento de estos diferentes resultados (es decir, la frecuencia de cada palabra) dentro de un documento que pertenece a una clase específica.

Distribución binomial

Una **distribución binomial** es una distribución de probabilidad discreta que se utiliza para modelar el número de éxitos en una serie de **ensayos de Bernoulli** independientes y repetidos. Un ensayo de Bernoulli es un experimento con solo dos resultados posibles: **éxito** o **fracaso** (por ejemplo, lanzar una moneda).

Para que una situación siga una distribución binomial, debe cumplir con las siguientes condiciones:

- **Número de ensayos fijo (n):** El experimento se repite un número determinado de veces.
- **Independencia:** El resultado de cada ensayo no afecta a los resultados de los otros.
- **Dos resultados posibles:** Cada ensayo solo puede tener dos resultados, éxito o fracaso.
- **Probabilidad de éxito constante (p):** La probabilidad de que ocurra un éxito es la misma en cada ensayo. La probabilidad de fracaso (q) es, por lo tanto, $1-p$.

La distribución binomial nos permite calcular la probabilidad de obtener un número específico de éxitos (k) en el total de ensayos (n).

Distribución multinomial

La **distribución multinomial** es una generalización de la distribución binomial. Mientras que la distribución binomial modela el resultado de una serie de ensayos con solo **dos resultados posibles** (como éxito o fracaso), la distribución multinomial modela el resultado de una serie de ensayos con **más de dos resultados posibles**.

Características Clave

La distribución multinomial se aplica a un experimento que cumple con las siguientes condiciones:

- Hay un número fijo de ensayos independientes (n).
- Cada ensayo tiene un resultado que pertenece a una de varias categorías (k), donde $k > 2$.
- La probabilidad de que un ensayo caiga en cada categoría (p_1, p_2, \dots, p_k) permanece constante a lo largo de los ensayos.

El principal objetivo de la distribución multinomial es calcular la probabilidad de obtener un número específico de resultados para cada una de las categorías después de realizar n ensayos.

Distribución multinomial: ejemplo

- Imagina un dado de seis caras que lanzas 10 veces. Cada lanzamiento es un ensayo. Los resultados posibles son 1, 2, 3, 4, 5 o 6 (hay 6 categorías). La distribución multinomial te permitiría calcular la probabilidad de obtener, por ejemplo, tres 1s, dos 2s, un 3, cero 4s, cuatro 5s y cero 6s.
- En el contexto de la clasificación de texto (como en el Naive Bayes Multinomial), las "categorías" son las palabras de un vocabulario, y los "ensayos" son las palabras que aparecen en un documento. La distribución multinomial modela la frecuencia con la que aparecen las palabras en el texto.

Clasificador bayesiano ingenuo multinomial

Ejemplos de uso:

- Filtro de spam
- Clasificación de noticias por tema
- Análisis de “sentimiento”

Clasificador bayesiano ingenuo multinomial

Ejemplo: Quiero clasificar una noticia según sea sobre China o Japón

	Doc.	Palabras	Clase
Entrenamiento	1	Chino Beijing Chino	China (c)
	2	Chino Chino Shanghái	China (c)
	3	Chino Macao	China (c)
	4	Tokio Japón Chino	Japón (j)
Prueba	5	Chino Chino Chino Tokio Japón	¿?

Clasificador bayesiano ingenuo multinomial

- Previos: la probabilidad de un documento de estar en cada clase

$$P(clase) = \frac{N_c}{N}$$

- Probabilidades condicionales de cada palabra:

$$P(palabra|clase) = \frac{\text{contar}(palabra, clase) + 1}{\text{contar}(todas las palabras, clase) + vocab.total}$$

Clasificador bayesiano ingenuo multinomial

Previos: $P(c) = \frac{3}{4}$ $P(j) = \frac{1}{4}$

Probabilidades condicionales:

- $P(\textit{Chino}|c) = \frac{(5+1)}{(8+6)} = 3/7$
- $P(\textit{Tokio}|c) = \frac{(0+1)}{(8+6)} = 1/14$
- $P(\textit{Japón}|c) = \frac{(0+1)}{(8+6)} = 1/14$
- $P(\textit{Chino}|j) = \frac{(1+1)}{(3+6)} = 2/9$
- $P(\textit{Tokio}|j) = \frac{(1+1)}{(3+6)} = 2/9$
- $P(\textit{Japón}|j) = \frac{(1+1)}{(3+6)} = 2/9$

Clasificador bayesiano ingenuo multinomial

Para el documento de prueba:

“Chino Chino Chino Tokio Japón”

$$\hat{y} = \operatorname{argmax} P(C_i) \prod P(\text{palabra}|\text{clase})$$

Elegimos una clase:

- $P(c|d5) \propto \frac{3}{4} * \left(\frac{3}{7}\right)^3 * \left(\frac{1}{14}\right)^2 = 0,0003$
- $P(j|d5) \propto \frac{1}{4} * \left(\frac{2}{9}\right)^3 * \left(\frac{2}{9}\right)^2 = 0,0001$

Se selecciona el máximo al comparar o sea: Chino.

Clasificador bayesiano ingenuo multinomial

Ejercicio: Usar los siguientes datos para entrenar el algoritmo

Doc.	Texto	Clase
1	I loved the movie	+
2	I hated the movie	-
3	A great movie. Good movie.	+
4	Poor acting	-
5	Great acting. A good movie.	+

Clasificador bayesiano ingenuo multinomial

- Obtener las palabras únicas
- Crear una tabla con las ocurrencias de cada palabra:

Nº doc.	Palabra 1	Palabra 2	...	Palabra N	Clase
1	1	0	...	5	+
2	2	1	...	1	-
...
N	3	2	...	2	+

Clasificador bayesiano ingenuo multinomial

- Indicar la clasificación para la siguiente frase:
 - “I hated the poor acting”
- Hacerlo con Excel y Python

Clasificador de SPAM

Naïve Bayes

Clasificador de SPAM

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

Expresar la probabilidad de que un correo sea spam si contiene la palabra “free”

$$P(spam|free) = \frac{P(spam)P(free|spam)}{P(free)}$$

Clasificador de SPAM

- Podemos construir $P(spam|word)$ por cada palabra que encontremos durante el entrenamiento.
- Luego se multiplican todas juntas para obtener la probabilidad de que un correo nuevo sea spam.
- Al ser “ingenuo” no tiene en cuenta las relaciones entre palabras.

Clasificador de SPAM

- Usaremos scikit-learn.
- CounterVectorizer nos deja operar con muchas palabras a la vez.
- MultinomialNB calcula las probabilidades.
- Haremos el entrenamiento con un listado de correos de los que se sabe que son spam o “ham”.



vs.



Clasificador de SPAM

- La mayor parte del trabajo es de limpiar y ordenar los datos para que puedan ser ingresados en la función de cálculo.
- Abrir: naivebayes.ipynb
- Actividades:
 - Probar correos spam nuevos y ver que resultado arroja
 - Hacer train/test para probar el modelo

Sparse matrix con CountVectorizer

CountVectorizer está diseñado para manejar colecciones de documentos de manera eficiente. La matriz resultante se llama a menudo una matriz de Término-Documento, donde:

- Filas: Representan cada uno de los documentos.
- Columnas: Representan cada una de las palabras únicas (términos) encontradas en todo el conjunto de datos.
- Valores: Son los conteos de frecuencia.

Dado que la mayoría de los documentos no contienen la mayoría de las palabras del vocabulario total, la matriz contendría muchos ceros. Por esta razón, el objeto devuelto es una matriz dispersa, que solo almacena los valores distintos de cero y sus ubicaciones, ahorrando una enorme cantidad de memoria.

NaiveBayes.ipynb

samples

Nº doc.	23667	54154	...	Palabra N
1	1	0	...	5
2	2	1	...	1
...
3000	3	2	...	2



```
<Compressed Sparse Row sparse matrix of dtype 'int64'  
  with 429785 stored elements and shape (3000, 62964)>
```

Coords	Values
(0, 20407)	1
(0, 28844)	5
(0, 44554)	1
(0, 57486)	1
(0, 21111)	1
(0, 54131)	1
(0, 22319)	1
(0, 27856)	2

(fila, columna) valor

NaiveBayes.ipynb

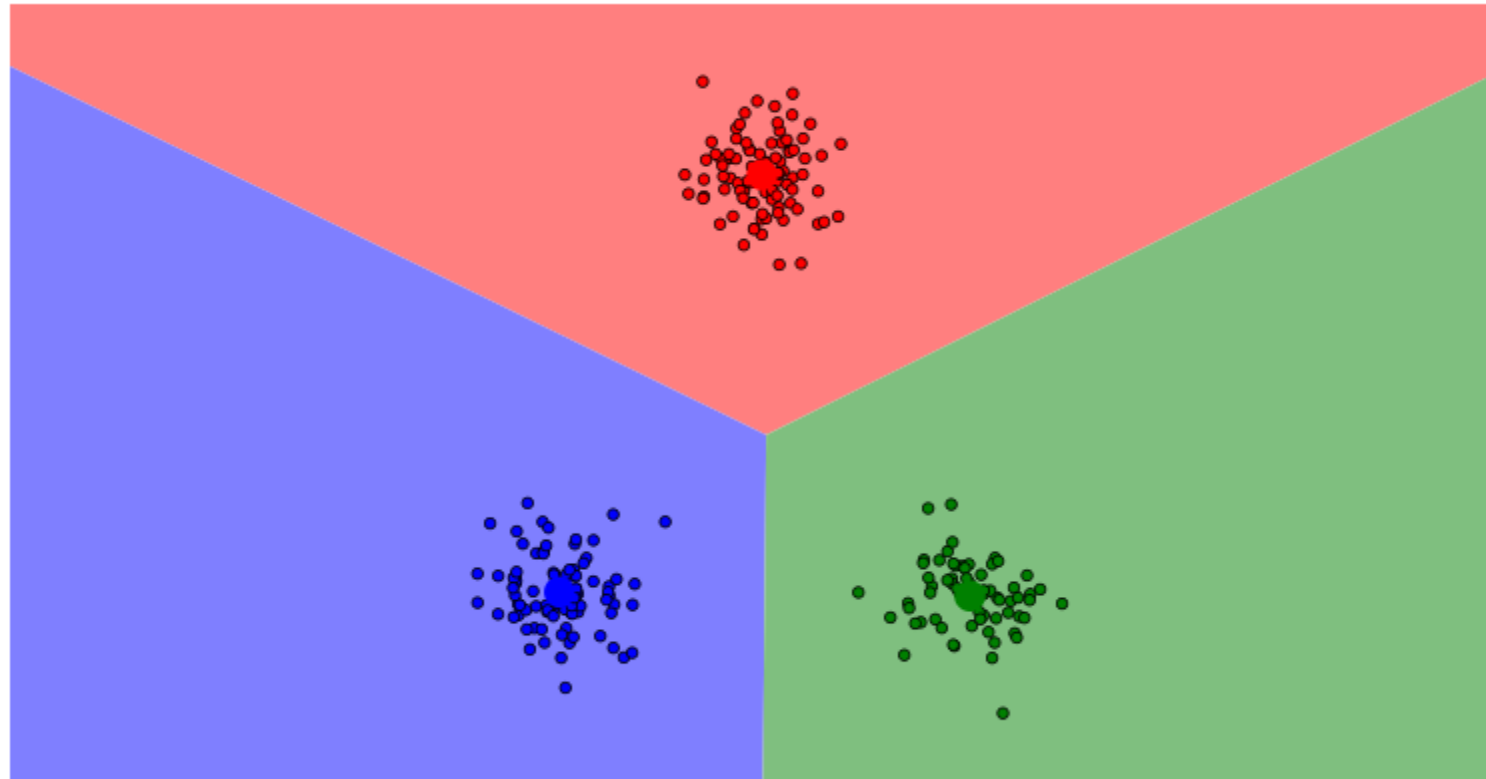
		counts (features)				targets
samples	Nº doc.	23667	54154	...	Palabra N	Clase
	1	1	0	...	5	spam
	2	2	1	...	1	spam

	3000	3	2	...	2	ham

K-means clustering

Agrupamiento K-Means

Demo



K-means clustering

- El **Algoritmo de k -Means** (o **k-Medias**) es un método de **aprendizaje automático no supervisado** muy popular, utilizado para resolver el problema de **agrupamiento** (*clustering*).
- Se utiliza para encontrar grupos naturales o patrones en un conjunto de datos donde las etiquetas o categorías son desconocidas de antemano.

Concepto y objetivo

- El objetivo de k-Means es particionar N puntos de datos en k grupos (o clusters) predefinidos, de tal manera que cada punto pertenezca al cluster cuyo centroide (media) le sea más cercano.
- En esencia, busca que:
 - Los puntos de datos dentro de un mismo cluster sean lo más similares posible entre sí.
 - Los grupos sean lo más diferentes posible entre ellos.



Funcionamiento

1. **Inicialización de k:** Se elige el **número k de clusters** que se desean formar.
2. **Inicialización de Centroides:** Se seleccionan k puntos **aleatorios** del conjunto de datos como los **centroides iniciales**.
3. **Medir la distancia** entre cada punto y los centroides.
4. **Asignación de Puntos (Clustering):** Cada **punto de datos se asigna al centroide más cercano** (generalmente utilizando la distancia euclidiana).
5. **Actualización de Centroides:** Se recalcula la posición de cada centroide, tomando la **media** de todos los puntos que le fueron asignados.
6. **Iteración:** Los pasos 3 y 4 se repiten hasta que:
 - Los centroides ya no cambien de posición.
 - Se alcance un número máximo de iteraciones.

Funcionamiento

Para el mismo k , se prueban distintos centroides aleatorios

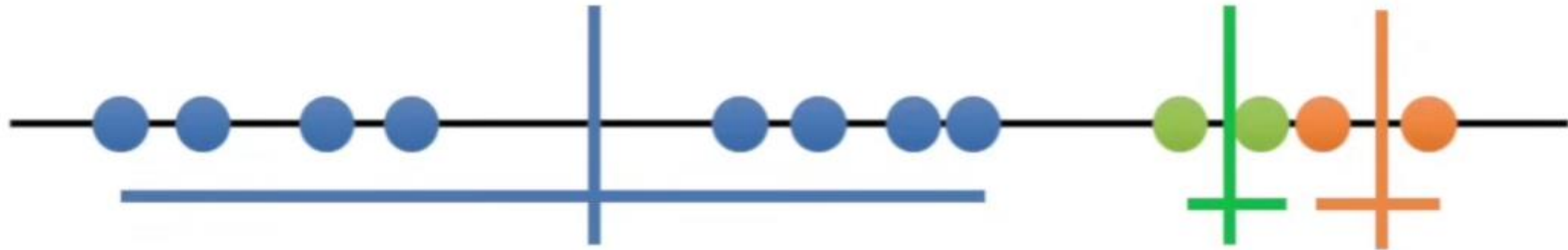
- Se elige el que tenga menos variación

Para distintos k , se elige el número óptimo con el método “elbow”

Funcionamiento paso a paso



Para distintos centroides con $k=3$



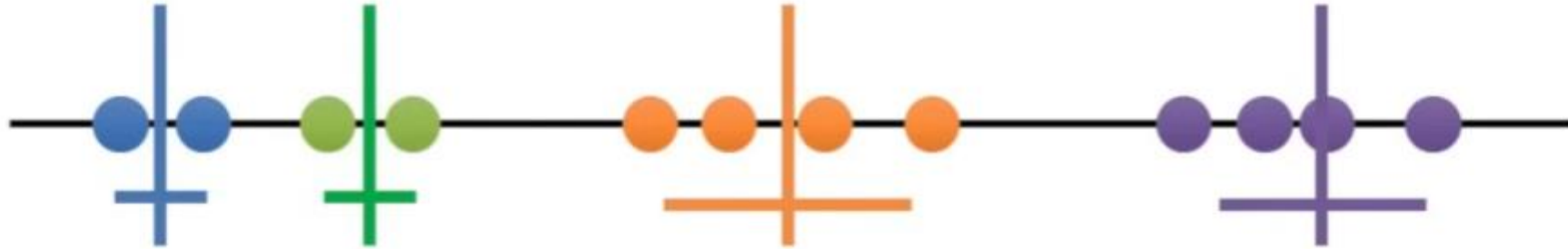
1st cluster attempt: 

2nd cluster attempt: 

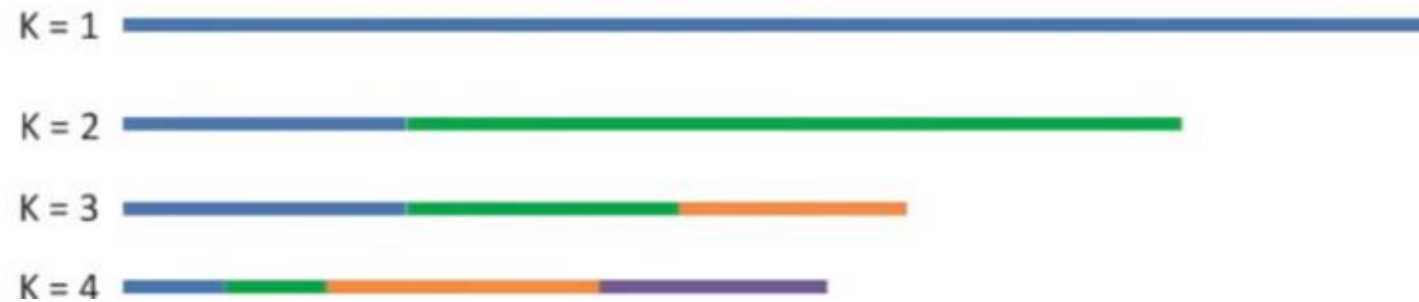
The winner!!

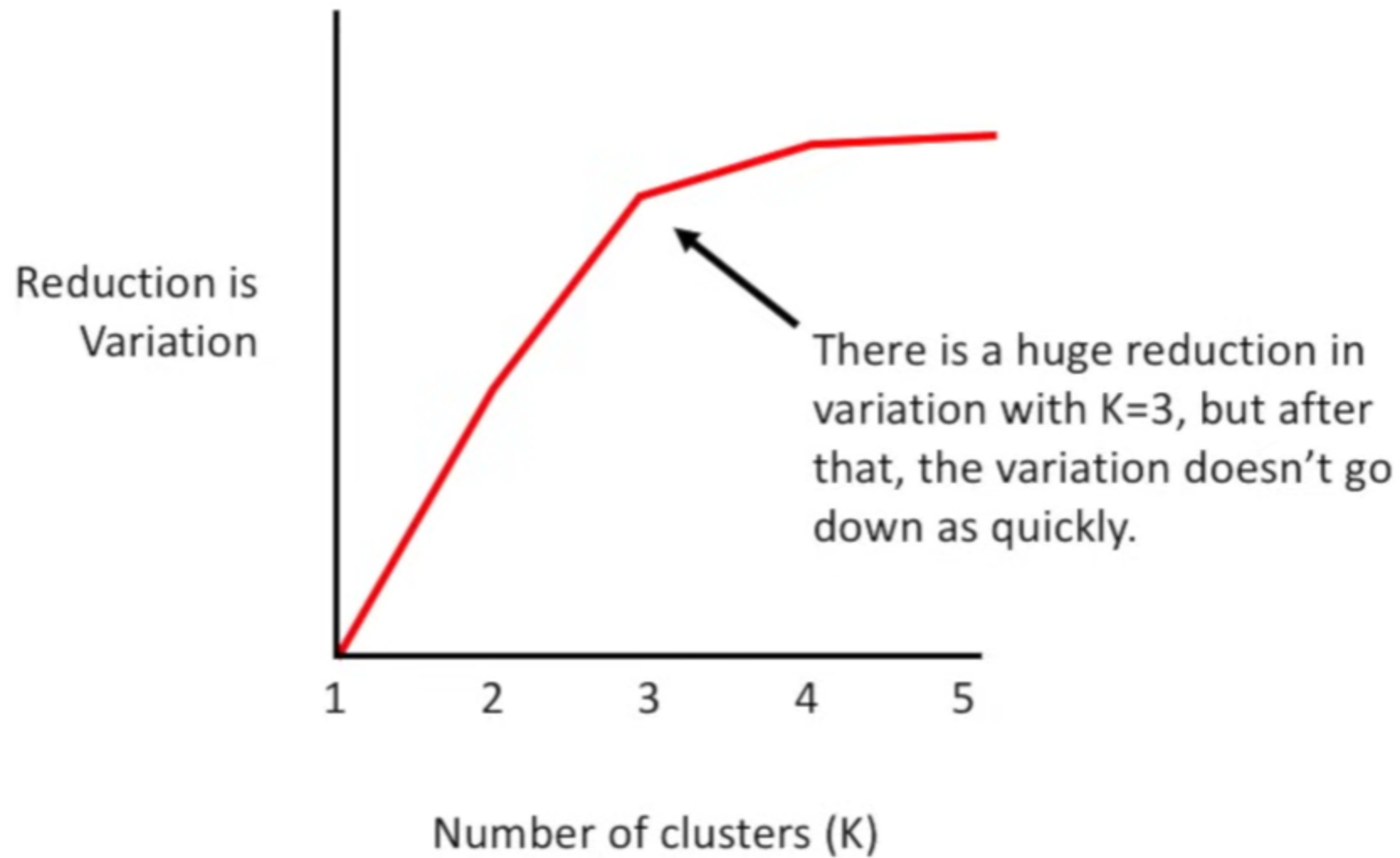
3rd cluster attempt: 

Para distintas k



The total variation within each cluster is less than when $K=3$





K-means clustering - A tener en cuenta

- K-Means no intenta asignar ningún significado a los clústeres que encuentra.
- Depende de ti profundizar en los datos e intentar determinar ese significado.
- **Sensibilidad a valores atípicos:** Los valores atípicos pueden influir desproporcionadamente en el cálculo del centroide, ya que el centroide es simplemente la **media** de todos los puntos de su grupo.
- Asume que los *clusters* son **esféricos, convexos** y de **tamaño similar**.

Ejemplo

- Abrir KMeans.ipynb

Ejercicio

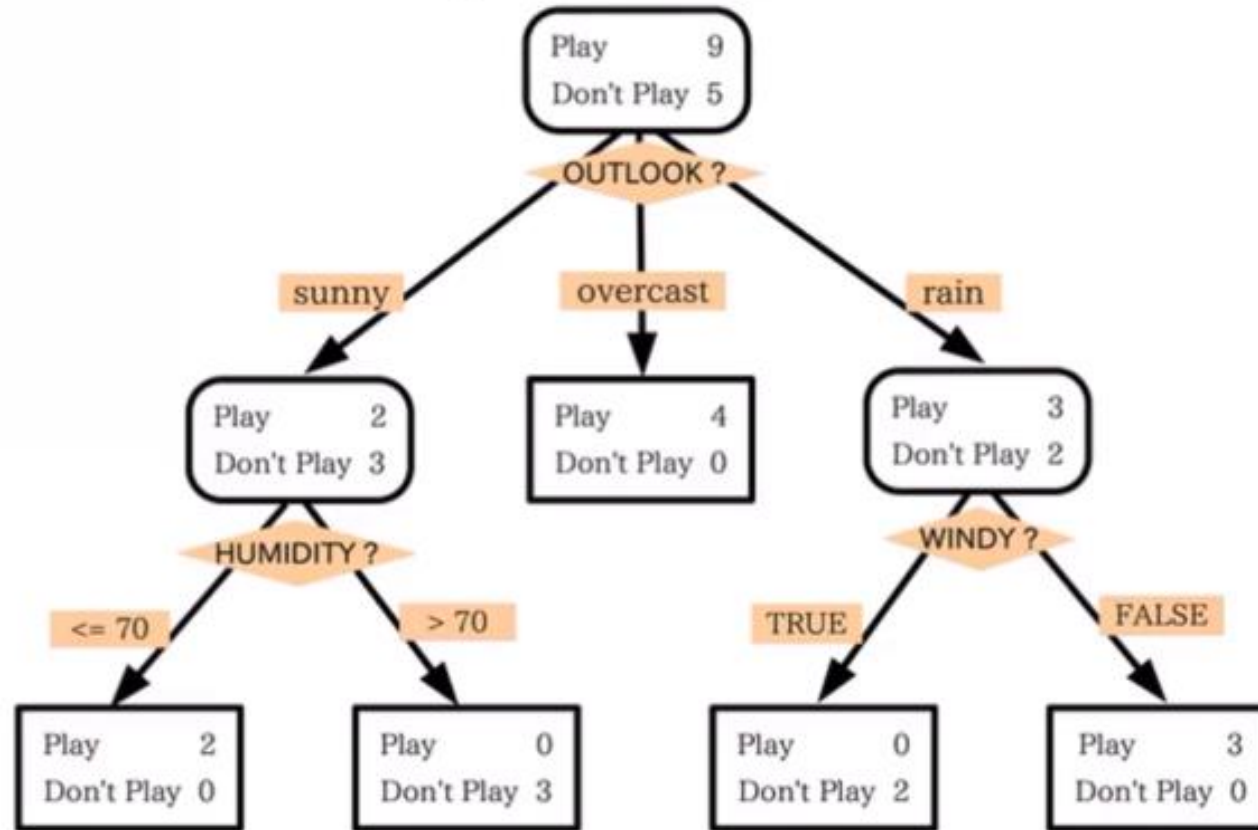
- Al ejercicio anterior añadir una forma de buscar el número de k óptimo
- Usar el dataset en Kaggle del FIFA 22 con estadísticas de los jugadores para agrupar los jugadores por ['overall', 'potential', 'wage_eur', 'value_eur', 'age']
- FIFA dataset

Decision trees

Definición

- Los **Árboles de Decisión** son un algoritmo de **aprendizaje supervisado** que se utiliza tanto para tareas de **clasificación** como de **regresión**. Toman su nombre y estructura de los diagramas de árbol, siendo muy valorados por su **fácil interpretación**.
- El modelo predice el valor de una variable objetivo (la variable a predecir, o *target*) aprendiendo reglas de decisión simples inferidas a partir de las características (atributos) de los datos.
- Nos da un diagrama que nos ayuda a tomar decisiones con los límites de cada una.

Dependent variable: PLAY



Ventajas

Los árboles de decisión son muy populares porque:

- **Fácil Interpretación (Explainable AI):** Su estructura visual tipo diagrama de flujo permite entender cómo se llegó a una predicción.
- **No Requieren Normalización:** Pueden manejar datos numéricos y categóricos sin necesidad de escalamiento previo.
- **Manejo de Variables:** Pueden trabajar con variables de diferentes tipos (continuas, discretas, categóricas).

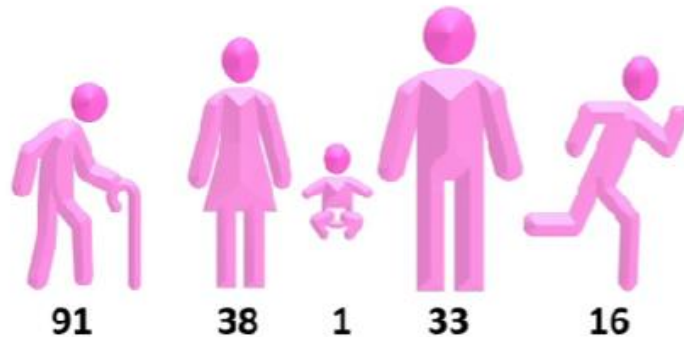
Desventajas

El principal problema de un único Árbol de Decisión es que es propenso al **sobreajuste** (*overfitting*) y tiene alta varianza. Por esta razón, en la práctica se usan con mayor frecuencia métodos de **ensamble** que combinan múltiples árboles, como **Random Forest** (Bosques Aleatorios) para mejorar la precisión y robustez.

La entropía

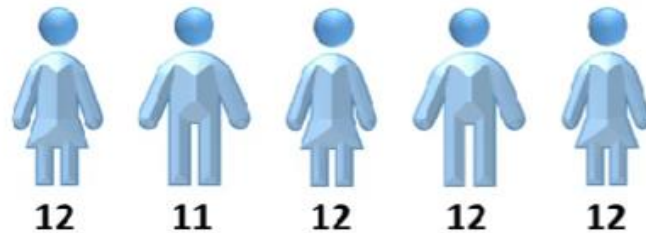
- Es una medida de la incertidumbre de una variable aleatoria
- La cantidad de información requerida para describir una variable

Edad
población
general



ALTA
Incertidumbre

Edad
niños de
6° de primaria



BAJA
Incertidumbre

How Decision Trees Work

- At each step, find the attribute we can use to partition the data set to minimize the *entropy* of the data at the next step
- Fancy term for this simple algorithm: ID3
- It is a *greedy algorithm* – as it goes down the tree, it just picks the decision that reduce entropy the most at that stage.
 - That might not actually result in an optimal tree.
 - But it works.

Random Forests

- Decision trees are very susceptible to overfitting
- To fight this, we can construct several alternate decision trees and let them “vote” on the final classification
 - Randomly re-sample the input data for each tree (fancy term for this: *bootstrap aggregating* or *bagging*)
 - Randomize a subset of the attributes each step is allowed to choose from

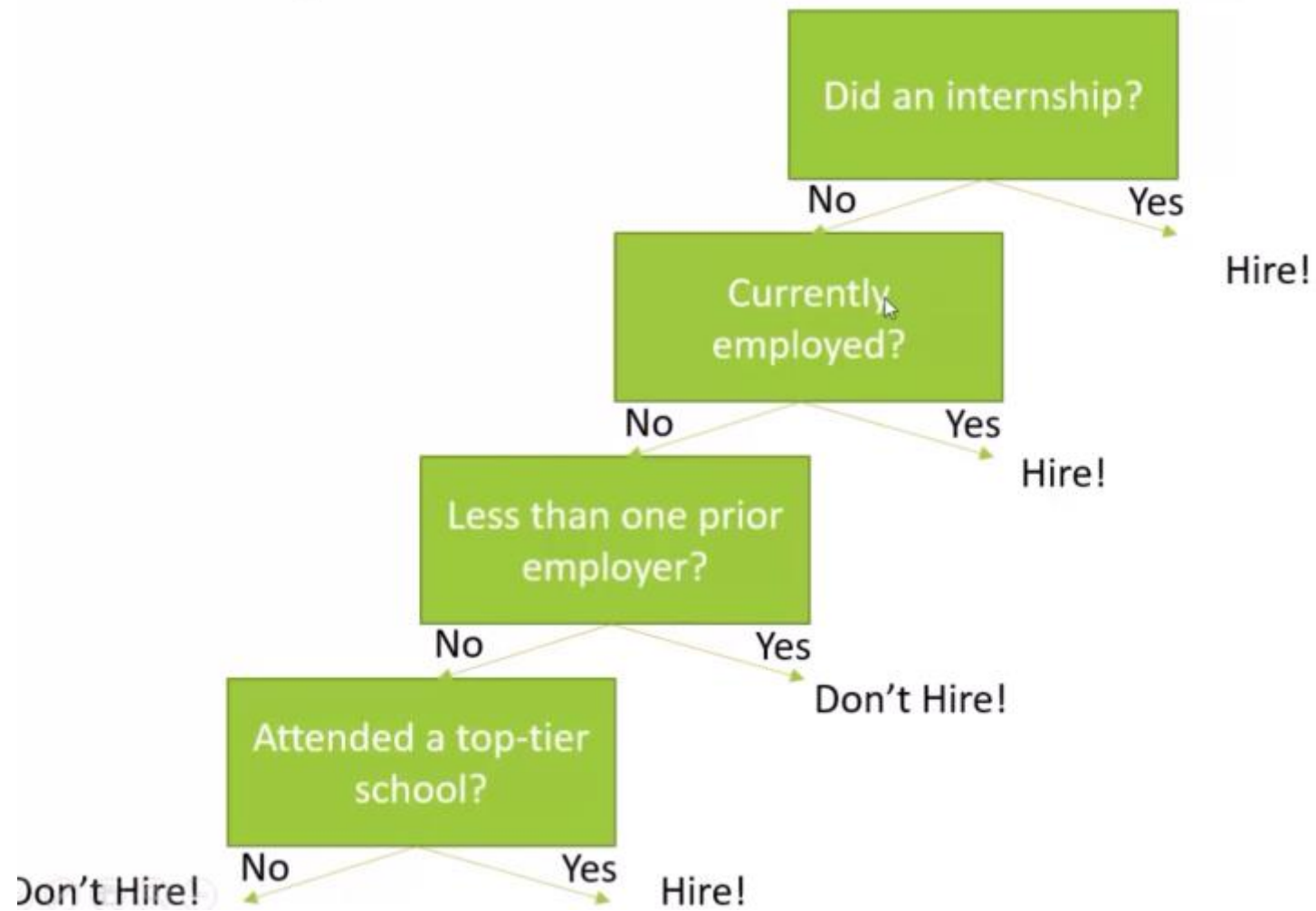
Ejemplo

- Queremos construir un sistema para filtrar currículums basado en datos históricos de contratación.
- Tenemos una base de datos con algunos atributos importantes de los candidatos a un puesto de trabajo, y sabemos cuáles fueron contratados y cuáles no lo fueron.
- Podemos entrenar un árbol de decisión con estos datos y obtener un sistema para predecir si un candidato será contratado basándote en ellos.
- Descargar `PastHires.csv` y `DecisionTree.ipynb`

El dataset



Candidate ID	Years Experience	Employed?	Previous employers	Level of Education	Top-tier school	Interned	Hired
0	10	1	4	0	0	0	1
1	0	0	0	0	1	1	1
2	7	0	6	0	0	0	0
3	2	1	1	1	1	0	1
4	20	0	2	2	1	0	0



Ejercicio