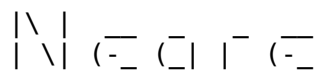
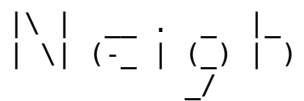
"Análisis de Algoritmos"
José Helo Guzmán





Contenido

▬▬▬▬ 1. Introducción.

En este capítulo aprenderemos como se construye un algoritmo de clasificación utilizando el método de KNN, K-nearest-neighbors.

Los principales elementos de KNN son, el mecanismo de distancia y extracción.

Una vez hecho el proceso de clasificación, se construirá un método de regresión. Regresión significa predecir. En nuestro caso trataremos de predecir un número. Este número puede estar asociado a el valor de una acción de la bolsa de valores en el futuro, o la cantidad de lluvia que caerá mañana.

Finalmente, veremos los casos en que se utiliza y algunas de sus limitaciones.

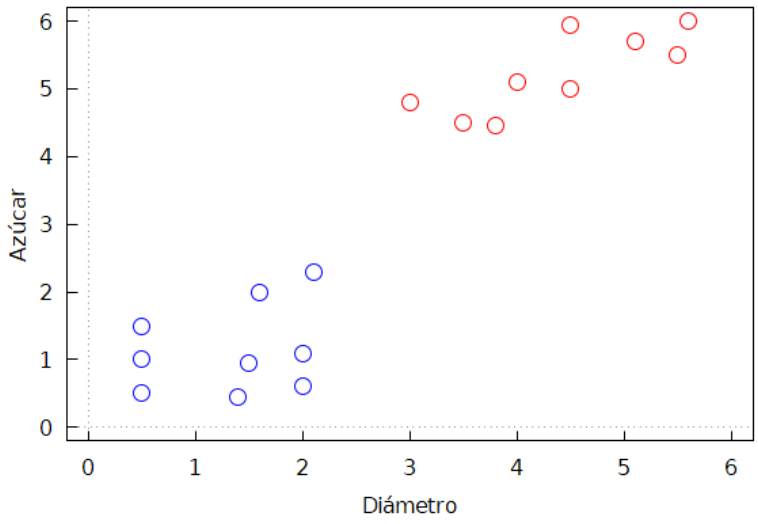
▬▬▬ 2. Clasificando Naranjas y Toronjas.

Supongamos que tenemos varias naranjas y toronjas. Las deseamos clasificar con base en dos atributos. Su tamaño, medido por el diámetro, y su nivel de dulzura. De esta forma cuando las clasificamos obtenemos el siguiente gráfico.

» Gráfico.
» Naranjas y Toronjas.

```
load(draw)$

naranjas:
[ [0.50, 0.50],
  [1.40, 0.45],
  [2.00, 0.60],
  [0.50, 1.00],
  [1.50, 0.95],
  [2.00, 1.10],
  [0.50, 1.50],
  [1.60, 2.00],
  [2.10, 2.30]
];


toronjas:
[ [3.50, 4.50],
  [3.80, 4.45],
  [3.00, 4.80],
  [4.50, 5.00],
  [4.50, 5.95],
  [4.00, 5.10],
  [5.50, 5.50],
  [5.60, 6.00],
  [5.10, 5.70]
];


ini:0;
fin:6;
wxdraw2d
(
   xrange = [ini-0.20,fin+0.20],
   yrange = [ini-0.20,fin+0.20],
   grid=false,
   xaxis=true,
   yaxis=true,
   xlabel="DiAmetro",
   ylabel="AzUcar",

   point_type    = 6,
   point_size    = 2,
   points_joined = false,

   color=blue,
   points(naranjas),

   color=red,
   points(toronjas)
```

)$

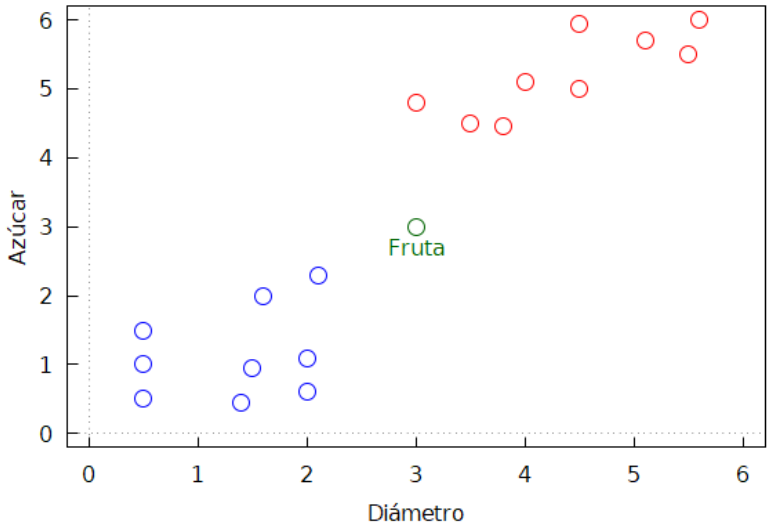


En general, podemos notar que entre más grande la fruta y con más azúcar o dulzura, entonces se trata de una toronja. Pero qué ocurre si tenemos que clasificar esta fruta?

» Gráfico.
» Fruta para Clasificar.

```
load(draw)$

naranjas:
[ [0.50, 0.50],
  [1.40, 0.45],
  [2.00, 0.60],
  [0.50, 1.00],
  [1.50, 0.95],
  [2.00, 1.10],
  [0.50, 1.50],
  [1.60, 2.00],
  [2.10, 2.30]
];


toronjas:
[ [3.50, 4.50],
  [3.80, 4.45],
  [3.00, 4.80],
  [4.50, 5.00],
  [4.50, 5.95],
  [4.00, 5.10],
  [5.50, 5.50],
  [5.60, 6.00],
  [5.10, 5.70]
];

misterio:
[ [3.00,3.00]
],


ini:0;
fin:6;
wxdraw2d
```

```
(
   xrange = [ini-0.20,fin+0.20],
   yrange = [ini-0.20,fin+0.20],
   grid=false,
   xaxis=true,
   yaxis=true,
   xlabel="DiAmetro",
   ylabel="AzUcar",

   point_type    = 6,
   point_size    = 2,
   points_joined = false,

   color=blue,
   points(naranjas),

   color=red,
   points(toronjas),

   color=darkgreen,
   points(misterio),
   label(["Fruta",3.00,2.70])

)$
```

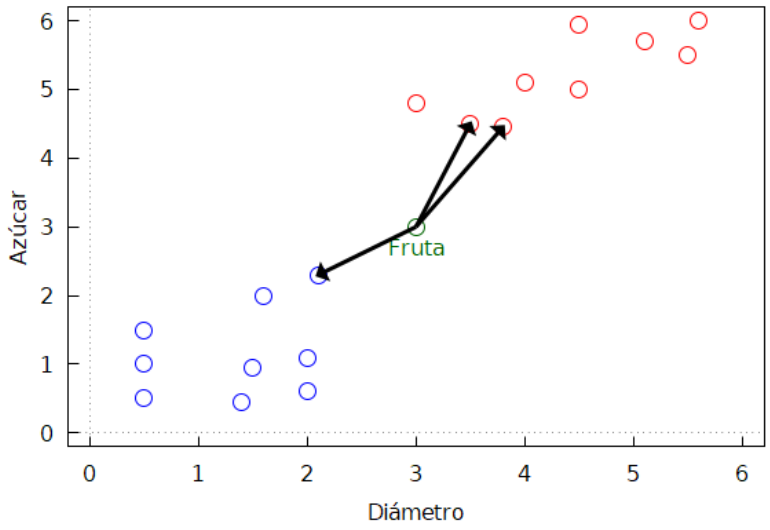


Primero encontramos las distancias más cercanas con “k” elementos.
En este caso usaremos k=3.

» Gráfico.
» Calculando Distancias.

```
load(draw)$

naranjas:
[
  [0.50, 0.50],
  [0.50, 1.00],
  [0.50, 1.50],
  [1.40, 0.45],
  [1.50, 0.95],
  [1.60, 2.00],
  [2.00, 0.60],
  [2.00, 1.10],
  [2.10, 2.30]
];
```

```
toronjas:
[
  [3.00, 4.80],
  [3.50, 4.50],
  [3.80, 4.45],
  [4.00, 5.10],
  [4.50, 5.00],
  [4.50, 5.95],
  [5.10, 5.70],
  [5.50, 5.50],
  [5.60, 6.00]
];

misterio:
[ [3.00,3.00]
];


ini:0;
fin:6;
wxdraw2d
(
   xrange = [ini-0.20,fin+0.20],
   yrange = [ini-0.20,fin+0.20],
   grid=false,
   xaxis=true,
   yaxis=true,
   xlabel="DiAmetro",
   ylabel="AzUcar",

   point_type    = 6,
   point_size    = 2,
   points_joined = false,

   color=blue,
   points(naranjas),

   color=red,
   points(toronjas),

   color=darkgreen,
   line_width=3,
   points(misterio),
   label(["Fruta",3.00,2.70]),

   line_width=3,
   head_length = 0.08,
   color       = black,
   line_type   = solid,
   vector([3.00,3.00],[0.50,1.50]),
   vector([3.00,3.00],[0.80,1.45]),
   vector([3.00,3.00],[-0.90,-0.70])
)$
```



En este ejemplo se puede notar como la mayoría de las frutas cercanas son toronjas. Así que lo más probable es que el nuevo elemento sea una toronja.

En general el proceso será el siguiente:

- Se tiene una nueva fruta que clasificar.
- Se revisan los vecinos más cercanos.
- Si más de los vecinos son naranjas, probablemente es una naranja.
  Si más de los vecinos son toronjas, probablemente es una toronja.

▬▬▬ 3. Medición de la Distancia.

Cómo se hace la medición para saber qué elementos son los vecinos más cercanos. Para eso debemos utilizar una fórmula de distancia.

En el ejemplo de naranjas y toronjas, utilizamos dos mediciones. El tamaño del diámetro y su nivel de dulzura. Supongamos que tenemos los elementos A y B, deseamos saber cuál de ellos es más cercano a C.

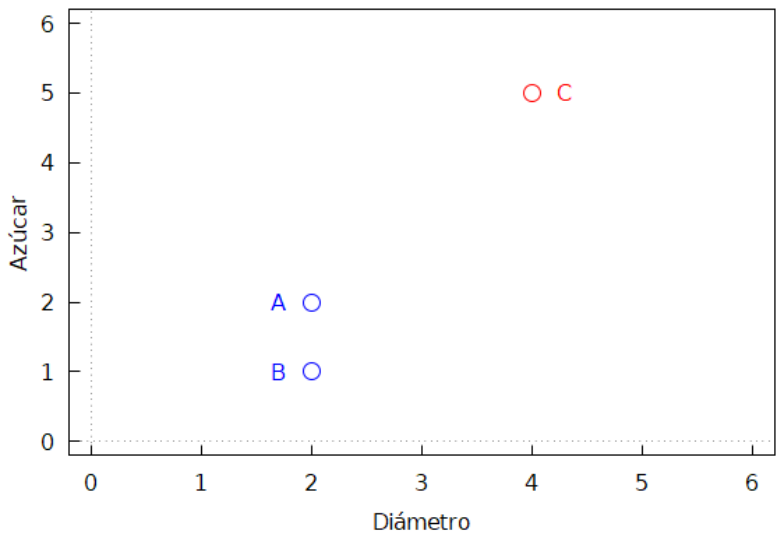
Aquí están las características de cada uno de los elementos.

	A	B	C
Diámetro:	2	2	4
Azúcar:	2	1	5

A continuación se muestra la localización de los puntos:

» Gráfico.

```
» Tres Puntos de Ejemplo

load(draw)$

a:
[ [2.00, 2.00]
];

b:
[ [2.00, 1.00]
];

c:
[ [4.00, 5.00]
];

ini:0;
fin:6;
wxdraw2d
(
   xrange = [ini-0.20,fin+0.20],
   yrange = [ini-0.20,fin+0.20],
   grid=false,
   xaxis=true,
   yaxis=true,
   xlabel="DiAmetro",
   ylabel="AzUcar",

   point_type    = 6,
   point_size    = 2,
   points_joined = false,

   color=blue,
   points(a),
   label(["A",1.70, 2.00]),
   points(b),
   label(["B",1.70, 1.00]),

   color=red,
   points(c),
   label(["C",4.30, 5.00])

)$
```



Para encontrar la distancia entre dos puntos se utiliza la siguiente fórmula:

$a = (a_1, a_2)$

$b = (b_1, b_2)$

Denominemos d(a,b) = distancia(a,b)

$$d(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$$

d(a,b) = sqrt($(a_1 - b_1)^2 + (a_2 - b_2)^2$)

●● Ejemplo de Medición de distancia:

Tenemos los siguientes puntos. El primer número indica el diámetro, el segundo número el nivel de azúcar.

A = (2, 2)

B = (2, 1)

C = (4, 5)

La distancia de A a C está dada por:

d(A,C) = sqrt($(2 - 4)^2 + (2 - 5)^2$)

d(A,C) = sqrt($(2 - 4)^2 + (2 - 5)^2$)

d(A,C) = 3.605551275463989

La distancia de B a C está dada por:

d(B,C) = sqrt($(2 - 4)^2 + (1 - 5)^2$)

d(B,C) = sqrt($(2 - 4)^2 + (1 - 5)^2$)

d(B,C) = 4.47213595499958

La distancia de A a B está dada por:

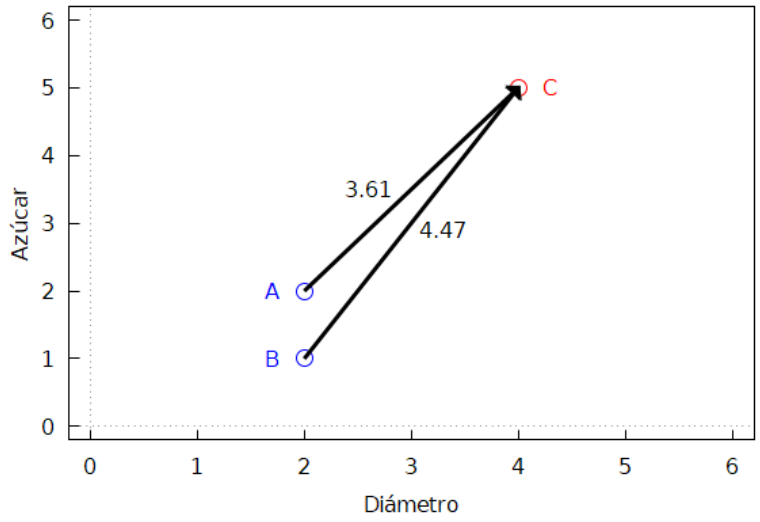
d(A,B) = sqrt($(2 - 2)^2 + (2 - 1)^2$)

d(A,B) = sqrt($(2 - 2)^2 + (2 - 1)^2$)

d(A,B) = 1

En el siguiente gráfico se pueden apreciar estas distancias:

```
» Gráfico.
» Calculando Distancias.

load(draw)$

a:
[ [2.00, 2.00]
];

b:
[ [2.00, 1.00]
];

c:
[ [4.00, 5.00]
];

ini:0;
fin:6;
wxdraw2d
(
    xrange = [ini-0.20,fin+0.20],
    yrange = [ini-0.20,fin+0.20],
    grid=false,
    xaxis=true,
    yaxis=true,
    xlabel="DiAmetro",
    ylabel="AzUcar",

    point_type    = 6,
    point_size    = 2,
    points_joined = false,

    color=blue,
    points(a),
    label(["A",1.70, 2.00]),
    points(b),
    label(["B",1.70, 1.00]),

    color=red,
    points(c),
    label(["C",4.30, 5.00]),

    line_width=3,
    head_length = 0.08,
    color       = black,
    line_type   = solid,
    vector([2,2],[2,3]),
    vector([2,1],[2,4]),
    label(["3.61",2.60,3.50]),
    label(["4.47",3.30,2.90])


)$
```



●● Distancia para Dimensiones Mayores.

Si tenemos dos puntos con “n” elementos:

$a = (a_1, a_2, a_3, \ldots, a_n)$

$b = (b_1, b_2, b_3, \ldots, b_n)$

Su distancia d(a,b) estará dada por:

$$d(a,b) = \sqrt{(a_1-b_1)^2 + (a_2-b_2)^2 + (a_3-b_3)^2 + \ldots + (a_n- b_n)^2}$$

$$d(a,b) = \text{sqrt}((a_1-b_1)^2 + (a_2-b_2)^2 + (a_3-b_3)^2 + \ldots + (a_n- b_n)^2)$$

●● Ejemplo.

Tenemos:

A = (3, 4, 4, 1, 4)

B = (4, 3, 5, 1, 5)

La distancia está dada por:

$d(A,B) = \text{sqrt}((3-4)^2 + (4-3)^2 + (4-5)^2 + (1-1)^2 + (4-5)^2)$

d(A,B) = sqrt((3-4)^2 + (4-3)^2 + (4-5)^2 + (1-1)^2 + (4-5)^2)

d(A,B) = 2

▬▬▬ 4. Un Sistema de Recomendaciones.

Supongamos que Netflix desea construir un sistema para la recomendación de películas para sus usuarios. De forma general, este problema es muy similar al problema anterior y se puede utilizar KNN.

Se recoleta información de los usuarios, esta información se puede analizar. Si se desea recomendar alguna película al usuario P, buscamos los usuarios con mayor cercanía a P. Posteriormente le recomendamos lo que han visto estos usuarios.

El primer paso consiste en modelar cada usuario como un conjunto de datos. Para ello tomamos las calificaciones de 1 a 5, que cada uno de estos usuarios le dio a los diferentes tipos de películas. Por ejemplo

	A	J	M
Comedia	3	4	2
Acción	4	3	5
Drama	4	5	1
Horror	1	1	3
Romance	4	5	1

Se desea hacer una recomendación para A. Cuál de los otros usuarios está mas cerca?

●● Calculando Distancias.

Para resolver el problema, tenemos que calcular diferentes distancias.

La distancia entre A y J está dada por:

$d(A,J) = \sqrt{(3-4)^2 + (4-3)^2 + (4-5)^2 + (1-1)^2 + (4-5)^2}$

d(A,J) = sqrt((3-4)^2 + (4-3)^2 + (4-5)^2 + (1-1)^2 + (4-5)^2)

d(A,J) = 2

La distancia entre A y M está dada por:

$d(A,M) = \sqrt{(3-2)^2 + (4-5)^2 + (4-1)^2 + (1-3)^2 + (4-1)^2}$

d(A,M) = sqrt((3-2)^2 + (4-5)^2 + (4-1)^2 + (1-3)^2 + (4-1)^2)

d(A,M) = 4.898979485566356

Por lo tanto se le recomendaría a A, ver las películas que ve J.

Por esta razón es que sistemas como Netflix constantemente le dicen que ponga el rating de más películas. Entre más películas califiquen los usuarios. Mejor serán las recomendaciones que se brindan para otros usuarios.

▬▬▬ 5. Regresión.

Las dos cosas básicas para las que sirve KNN son para:

• Clasificación, poner un elemento en un grupo o categoría.

• Regresión, predecir la respuesta, usualmente un número de una nueva entrada.

La regresión es muy útil. Veamos un ejemplo. Supongamos que tenemos una panadería. Se tienen que cocinar los panes todos los días en la mañana. Deseamos predecir el número de panes que se deben preparar el día de hoy. Para ello contamos con los siguientes datos:

• El clima en escala de 1 a 5. (C)
  1 es un clima malo, un día lluvioso y oscuro.
  5 es un clima bueno, un día soleado y agradable.

• Es un día de fin de semana o feriado. (F)
  1 es un día de fin de semana o feriado.
  0 en otro caso.

• Hay algún partido de fútbol? (P)
  1 si hay partido.
  0 no hay partido.

Tenemos además la información de varios días.

Día	C	F	P	Cantidad de Panes
A	5	1	0	300
B	3	1	1	225
C	1	1	0	75
D	3	0	1	200
E	4	0	0	150
F	2	0	0	50

Vamos a guardar los datos de la siguiente forma:

datos:

```
[ [A, [5, 1, 0, 300]],
  [B, [3, 1, 1, 225]],
  [C, [1, 1, 0,  75]],
  [D, [3, 0, 1, 200]],
  [E, [4, 0, 0, 150]],
  [F, [2, 0, 0,  50]]
];
```

El día de hoy, tenemos buen clima, es un fin de semana y no hay partido de fútbol. Por lo que podemos categorizar el día como:

```
X = [4, 1, 0]
```

Para poder realizar la regresión, primero sacamos las distancias con todos los elementos.

Distancia entre A y X:

```
(%i) distancia(
     [5, 1, 0],
     [4, 1, 0]
     );
(%o) 1.0
```

Distancia entre B y X:

```
(%i) distancia(
     [3, 1, 1],
     [4, 1, 0]
     );
(%o) 1.414213562373095
```

Distancia entre C y X:

```
(%i) distancia(
     [1, 1, 0],
     [4, 1, 0]
     );
(%o) 3.0
```

Distancia entre D y X:

```
(%i) distancia(
     [3, 0, 1],
     [4, 1, 0]
     );
(%o) 1.732050807568877
```

Distancia entre E y X:

```
(%i) distancia(
     [4, 0, 0],
     [4, 1, 0]
     );
(%o) 1.0
```

Distancia entre F y X:

```
(%i) distancia(
     [2, 0, 0],
     [4, 1, 0]
     );
(%o) 2.23606797749979
```

Distancia de “X” con cada uno de los elementos de la base de datos.

Distancia:

A: 1.0

B: 1.414213562373095

C: 3.0

D: 1.732050807568877

E: 1.0

F: 2.23606797749979

Si suponemos que escogemos k=3 elementos tendremos los elementos A,B,E.

A: 1.0

E: 1.0

B: 1.414213562373095

Para obtener la respuesta del problema, podemos sacar un promedio de la cantidad de panes que se produjeron con estos elementos

A: 300 panes

E: 150 panes

B: 225 panes

$$X = \left(\frac{300 + 150 + 225}{3} \right)$$

X = 225 panes

▬▬▬ 6. Similitud del Coseno.

Hasta este momento siempre hemos utilizado la misma fórmula de distancia entre los puntos. Esta forma de medir la distancia se conoce como distancia euclidiana. Existen otras formas de medir la distancia.

Una de estas formas es la similitud del coseno. Esta distancia no mide la distancia entre dos vectores, sino que compara los ángulos de los vectores. Qué significa esto.

Suponga que un usuario A siempre valora todas las cosas que le gustan con una nota de 5. Mientras que el usuario B siempre valora las cosas que le gustan con una nota de 4, nunca da un puntaje de 5.

Estos dos usuarios pueden tener gustos parecidos aunque sus valoraciones difieran en un un punto. En estos casos es cuando se utiliza la similitud del coseno.

▬▬▬ 7. Normalización de los Datos.

En nuestros ejemplos los datos se encuentran en un intervalo relativamente cercano. Sin embargo podría ocurrir que una variable esté medida en centímetros y otra en millones. En ese caso, la diferencia de la segunda variable sería la que incide con mayor fuerza en la distancia.

	var1 en cm	var2 en millones
A	0.10	1000000

Si tenemos una medición de “X” dada por

X = [0.15, 950000]

Observe como el segundo elemento influye en la distancia:

d(A,X) = sqrt((0.10-0.15)^2 + (1000000-950000)^2)

d(A,X) = sqrt(0.05^2 + 50000^2)

d(A,X) = sqrt(0.0025 + 2500000000)

d(A,X) = sqrt(2500000000.0025)

d(A,X) = 50000.000000025

En estos casos es necesario normalizar los datos de entrada. Es usual que la normalización se de en un intervalo [0,1]. Existen muchas formas para normalizar los datos. A continuación mostramos un ejemplo sencillo.

En general si tenemos una variable “x” en un intervalo [a,b].

Se puede convertir al intervalo [0,1] utilizando la fórmula:

$$xNormalizado = \left(\frac{x - a}{b - a} \right)$$

●● Ejemplo

Si tenemos una variable var1, medida en el intervalo [1,5]. Hagamos algunas normalizaciones.

>>> Supongamos que tiene un valor de x=3, en el intervalo [1,5].

Podemos convertirla al intervalo [0,1] de la siguiente manera:

$$\left(\frac{3 - 1}{5 - 1} \right) = \left(\frac{2}{4} \right) = 0.50$$

>>> Supongamos que tiene un valor de x=1, en el intervalo [1,5].

Podemos convertirla al intervalo [0,1] de la siguiente manera:

$$\left(\frac{1 - 1}{5 - 1} \right) = \left(\frac{0}{4} \right) = 0.00$$

>>> Supongamos que tiene un valor de x=5, en el intervalo [1,5].

Podemos convertirla al intervalo [0,1] de la siguiente manera:

$$\left(\frac{5 - 1}{5 - 1} \right) = \left(\frac{4}{4} \right) = 1.00$$

▬▬▬ 8. Análisis de la Complejidad.

El algoritmo de KNN realiza los siguientes pasos, para categorizar un nuevo elemento “x”:

1. Sacar la distancia de “x” con todos los elementos.

2. Escoger los “k” vecinos más cercanos.

3. Hace alguna operación con los “k”vecinos para hacer
   una clasificación o una regresión.

(1) Sacar la distancia de “x” con todos los elementos.

Para este paso si tenemos “n” elementos y “m” variables. El tiempo de ejecución estará dado por:

$O(n*m)$

(2) Escoger los “k” vecinos más cercanos.

Para el siguiente paso se pueden tener una lista ordenada, o buscar linealmente los elementos más cercanos. Cualquiera que sea el caso, este tiempo es menor al anterior. Supongamos que usamos un Heap.

$O(n*\log_2(n))$

(3) Hace alguna operación con los “k”vecinos

Para el tercer paso, depende usualmente es una operación sencilla. En el peor de los casos se utilizarían todos los elementos, es decir k=n, por lo tanto el tiempo de ejecución sería:

$O(n)$

Por lo tanto el tiempo final está dado por:

$O(n*m) + O(n * \log_2(n)) + O(n)$

Que es equivalente a :

$O(n*m)$

▬▬▬ 9. Algunas Aplicaciones de KNN.

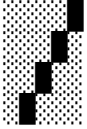
El algoritmo de KNN es muy útil y es muy utilizado en el campo de la inteligencia artificial y aprendizaje automático (“Machine Learning”).

Veamos algunos ejemplos de sus usos.

●● Optical Character Recognition (OCR)

Se trata de tomar una imagen, digitalizarla, y la computadora puede leer el texto. Google utiliza OCR para la digitalización de libros. Cómo funciona?

Supongamos que queremos hacer un reconocimiento de números. Observe este número que se tiene abajo.



Se puede representar como un vector de la forma:

```
numero: [ 1,1,1,1,1,
          0,0,0,0,1,
          0,0,0,1,0,
          0,0,1,0,0,
          0,1,0,0,0
         ];
```

O equivalentemente:

```
numero: [ 1,1,1,1,1,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0 ];
```

Si se tiene una base de datos con muchos números, cuando aparece un nuevo número, se debe establecer cuáles elementos de la base de datos son los más similares posibles.

●● Construcción de un Filtro de Spam.

Los filtros de Spam usan un algoritmo muy sencillo llamado “Naive Bayes Classifier”, clasificador ingenuo de Bayes. Primero se entrena el clasificador con algunos datos.

Frase	Resultado
“Cambie su password”	No Spam
“Ganó un millón de dólares”	Spam
“Quero regalarle un carro”	Spam
“Príncipe le envía 10 millones”	Spam
“Feliz Cumpleaños”	No Spam

El algoritmo descompone cada una de estas frases en palabras y calcula la probabilidad que una palabra está asociada con un email de spam. En este caso la palabra “millones” siempre está asociada a una comunicación de spam.

Para ello utiliza una modificación del algoritmo de KNN.

●● Predicción del Valor de Una Acción.

Las empresas venden acciones. Estas acciones tienen un valor presente. Se espera que en el futuro tengan un valor mayor, entonces se puede vender y así lograr una ganancia.

Si se tienen las características adecuadas, de cuales acciones han subido su valor, se puede utilizar KNN para clasificar una nueva acción que se desea comprar.

Desafortunadamente, este es uno de los sistemas más difíciles de construir, y hay muchas interferencias externas que lo hacen aún más difícil.

▬▬▬ 10. Resumen.

• KNN se puede utilizar para problemas de clasificación y para problemas de regresión.

• Se necesita siempre encontrar un grupo de “k” elementos que sean lo más cercanos posibles al caso de entrada.

• Es muy importante encontrar características significativas para que el algoritmo funcione correctamente.

• Además de la distancia euclideana, existen otras posibles distancias que se pueden utilizar.

• En algunos casos es necesario normalizar los valores de las variables.

• KNN tiene muchas aplicaciones. En especial en el campo de “machine learning”.

▬▬▬ 11. Ejercicios.

●● Ejercicio 1.

Se tiene el siguiente conjunto de usuarios.
Se desea hacer una recomendación para A.
Indique cuál de los otros usuarios está más cerca de A.

	A	B	C	D	E
Comedia	3	3	2	1	5
Acción	4	3	5	3	3
Drama	4	5	1	3	2
Horror	1	1	3	5	1
Romance	4	5	1	2	4

●● Ejercicio 2.

Netflix tiene un grupo de “influencers”. Por ejemplo, directores de cine como Quentin Tarantino y Wes Anderson son influencers en Netflix. De esta manera sus ratings tienen un valor mayor al de los usuarios normales.

Cómo puede cambiar el sistema de KNN para darle mayor peso a estos influencers?

●● Ejercicio 3.

Sigamos con el sistema de recomendación de películas. Puede ser que un usuario le haya dada una gran recomendación a la película, “La Vida de Helo”. Pero solo vio unos pocos minutos de esa película.

En cambio le dio una mala recomendación a la película “La Vida de Turing”, pero la ha visto varias veces.

Puede pensar en una forma de mejorar el sistema de recomendaciones?