Procedimiento de vecinos más cercanos con matrices de distancias parcialmente observadas

Aldo R. Franco Comas

Director: Andrés M. Alonso

Máster Universitario en Ingeniería Matemática.

Motivación

k-NN

- $\{(x_i, y_i)\}$ con x_i de longitud p.
- Sea x_0 un nuevo caso, se calculan las distancias $d(x_0, x_i)$ para todo $i = 1, ..., n$.
- Se buscan los k casos más cercanos.

k-NN Ventajas

- No paramétrico.
- Algoritmo simple.
- Alta precisión.
- Múltiples clases.
- Clasificación y regresión.
- Variedad de distancias.

KNN Desventajas

- Datos no balanceados.
- Computacionalmente costoso.
- Características homogéneas.
- Tratamiento con valores perdidos.

Objetivos

¿Qué se propone?

Un procedimiento k-NN donde no sea necesario calcular todas las distancias cada vez que tengamos que predecir un punto.
Supondremos que solo podemos calcular $(1-\ell)\%$ de dichas distancias.

¿Por qué?

Es posible que no sea factible calcular todas esas distancias:

- Tiempo de respuesta.
- Coste computacional.
- Pruebas destructivas.

Repositorio

`https://github.com/aldofranco91/TFM_Ing_Mat`

Definiciones básicas

- Distancia euclidiana: $d_2(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
- Desigualdad triangular: $m_{ik} \leq m_{ij} + m_{jk}$
- Desigualdad cuadrangular: $m_{ij} + m_{kl} \leq \max[m_{ik} + m_{jl}; m_{il} + m_{jk}]$
- Desigualdad ultramétrica: $m_{ij} \leq \max[m_{ik}; m_{jk}]$
- Error medio absoluto: $\frac{\sum_{i=1}^{k} \left| \widehat{O}_i - O_i \right|}{k}$
- Diferencia relativa entre matrices: $\frac{\|O - P\|_2}{\|O\|_2}$
- Diferencia relativa entre vectores: $\frac{\|o - p\|_2}{\|o\|_2}$
- Índice de Jaccard: $\frac{|A \cap B|}{|A \cup B|}$

Revisión de la literatura

Completamiento de matrices de distancias

- Problema de la métrica más cercana.
- Problema de la inferencia filogenética.

Relación entre el k-NN propuesto y el completamiento de matrices de distancias

- Asumimos que conocemos la matriz de distancia entre los n puntos de la muestra de entrenamiento.
- Creamos una nueva matriz de distancias $(n+1) \times (n+1)$, siendo la última fila/columna la correspondiente a x_0.
- Calculamos $(1-\ell)\%$ de las distancias de esa fila, las restantes las imputamos.

Problema de la métrica más cercana

Supongamos que tenemos una matriz $\boldsymbol{D}$ cuyos elementos deben cumplir las desigualdades triangulares pero en algunos casos no se verifican.

El problema de la métrica más cercana consiste en encontrar una matriz $\boldsymbol{M}$ cuyos elementos cumplan las desigualdades triangulares y que esté próxima a $\boldsymbol{D}$.

El algoritmo triangle fixing resuelve este problema para las metricas L_1, L_2 y L_∞.

Inicialización del algoritmo

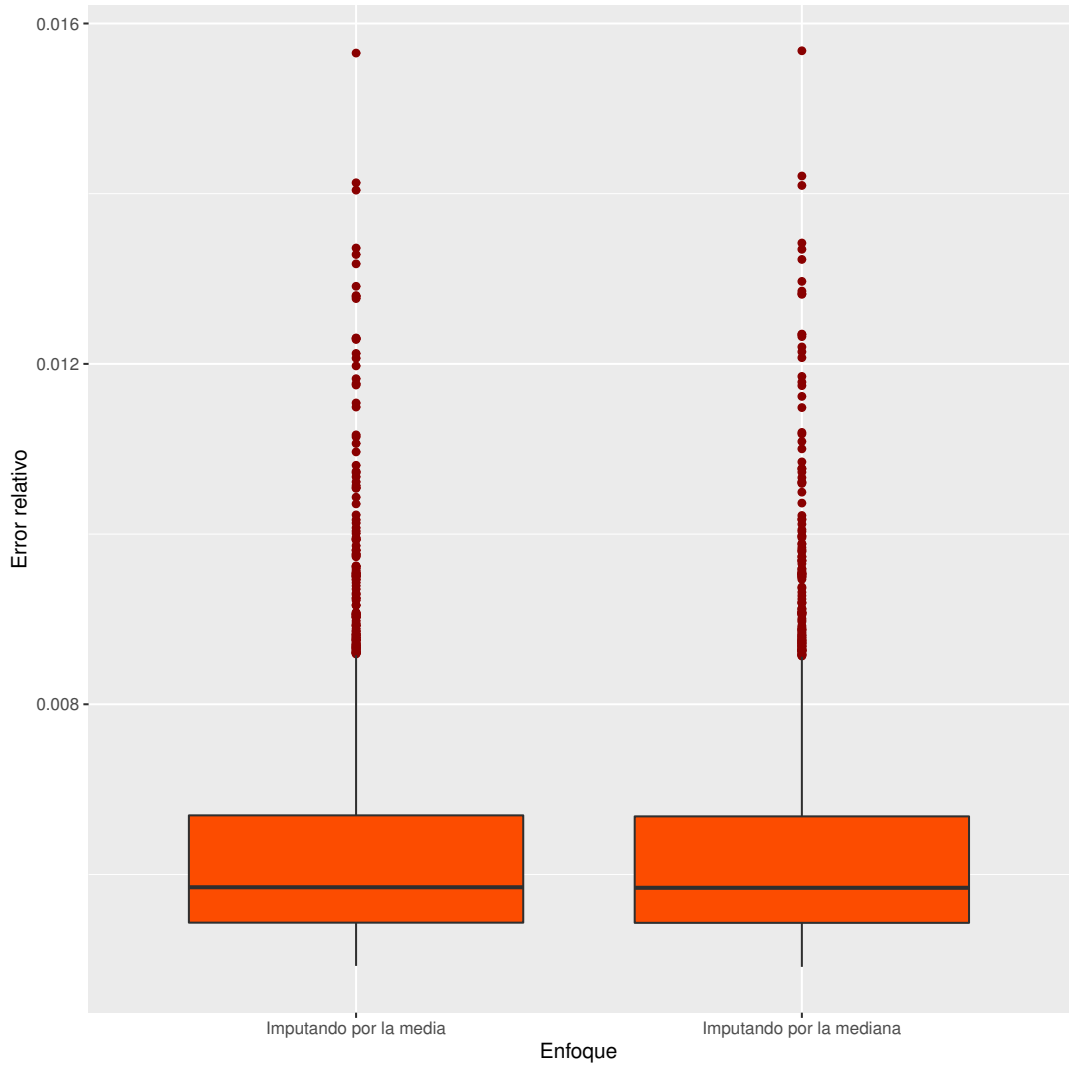
- $d(x_0, x_j) = 0$
- $d(x_0, x_j) = n^{-1} \sum_{i=1}^{n} d(x_i, x_j) \quad , i \neq j$
- $d(x_0, x_j) = \text{mediana}(d(x_i, x_j)) \quad , \forall i \neq j$

Problema de la métrica más cercana

Ejercicio de simulación

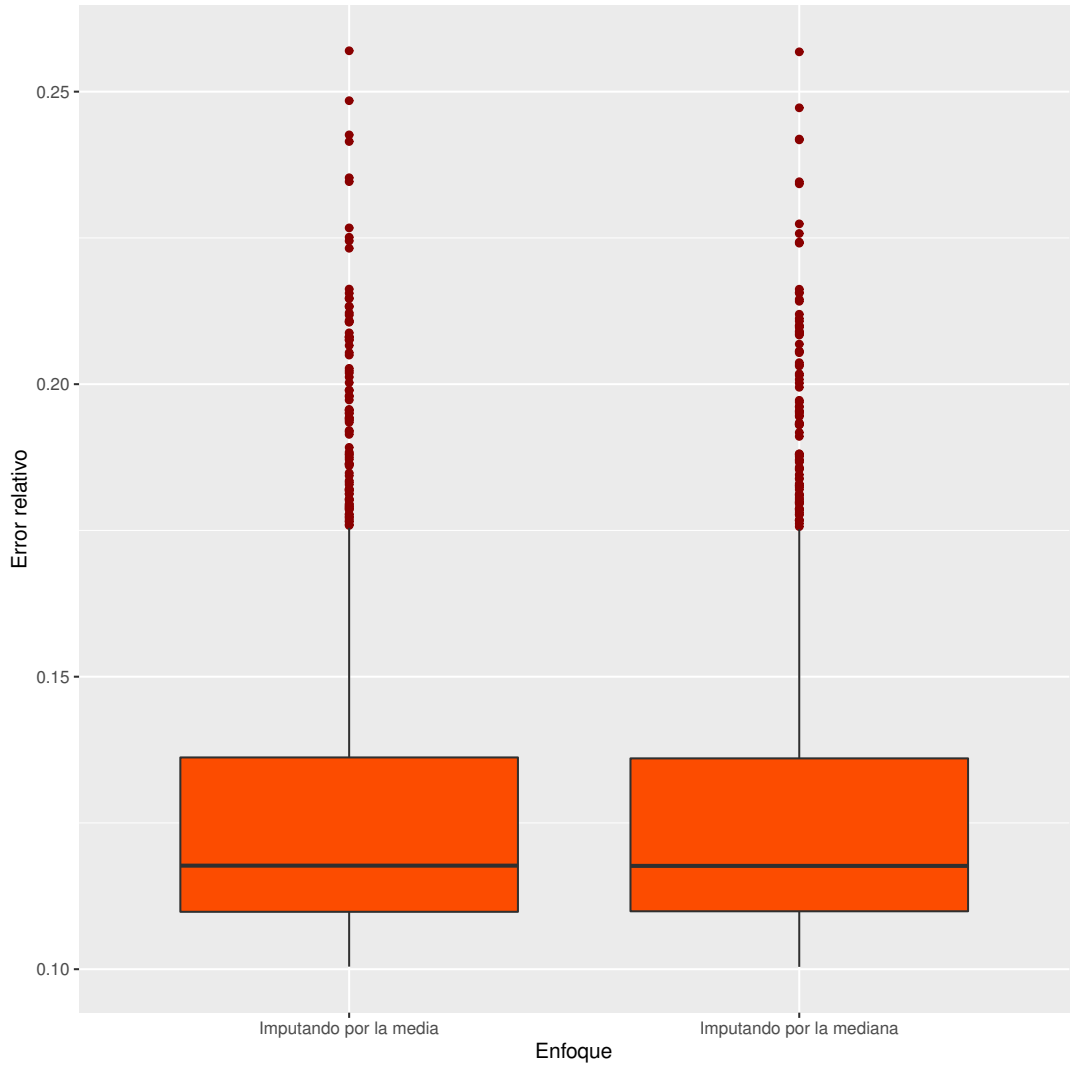
1. Se generan $n = 800$ puntos de una distribución normal multivariante con $\mu = \vec{0}_p$ y $\sigma = I_p$, siendo $p = 20$.
2. Se calcula la matriz de distancias, $\mathcal{D}$, usando la distancia euclidiana.
3. Se genera un nuevo punto x_0 y se calculan las n distancias.
4. Se asumen conocidas las distancias de x_0 a $n(1 - \ell) = 80$ puntos al azar y las restantes distancias se imputan usando las distintas inicializaciones, con lo cual tenemos una matriz $\boldsymbol{D}$.
5. Se aplica el triangle fixing a $\boldsymbol{D}$ y nos devuelve una matriz $\boldsymbol{M}$.
6. Se calcula la diferencia relativa entre la matriz $\boldsymbol{M}$ y la matriz $\mathcal{D}$ y la diferencia relativa entre el vector de distancias correspondiente a x_0 en la matriz $\mathcal{D}$ y en $\boldsymbol{M}$. Se registra el tiempo de cómputo.
7. Todo lo anterior se hace $N = 2000$ veces.

Problema de la métrica más cercana



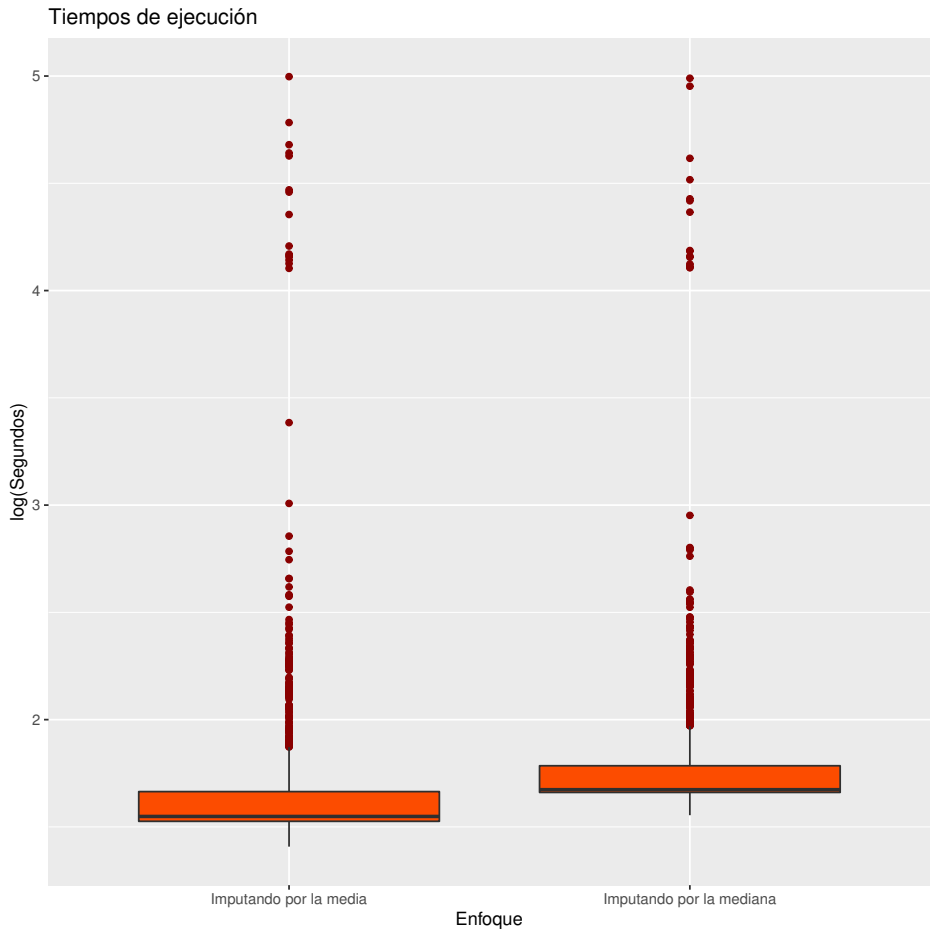
Diagramas de caja de las diferencias relativas entre $\mathcal{D}$ y $\boldsymbol{M}$

Problema de la métrica más cercana



Diagramas de caja de las diferencias relativas entre vectores de distancia.

Problema de la métrica más cercana



Diagramas de caja de los tiempos de computo.

Problema de la métrica más cercana: Método propuesto

Restricciones

- Las distancias $d(x_i, x_j)$ son conocidas $\forall i, j = \{1, ..., n\}$.
- Podemos calcular las distancias de $d(x_0, x_i)$ cuando $i \in I$ y $|I| << n$.
- No podemos calcular las $d(x_0, x_i)$ cuando $i \in I^c$.
- $|I| \approx (1 - \ell)n$ y $|I^c| \approx \ell n$.

Problema de la métrica más cercana: Método propuesto

Acotación

$$d(x_0, x_{i^*}) \leq d(x_0, x_i) + d(x_i, x_{i^*}) \quad \forall i^* \in I^c$$

$$d(x_0, x_{i^*}) \leq \min_{i \in I} \{d(x_0, x_i) + d(x_i, x_{i^*})\}$$

$$d(x_i, x_{i^*}) \leq d(x_0, x_{i^*}) + d(x_0, x_i) \quad \forall i^* \in I^c$$

$$d(x_0, x_i) \leq d(x_0, x_{i^*}) + d(x_i, x_{i^*}) \quad \forall i^* \in I^c$$

$$\max_{i \in I} |d(x_0, x_i) - d(x_i, x_{i^*})| \leq d(x_0, x_{i^*})$$

$$\max_{i \in I} |d(x_0, x_i) - d(x_i, x_{i^*})| \leq d(x_0, x_{i^*}) \leq \min_{i \in I} \{d(x_0, x_i) + d(x_i, x_{i^*})\}$$

$$d(x_0, x_{i^*}) = \frac{1}{2} \left(\min_{i \in I} \{d(x_0, x_i) + d(x_i, x_{i^*})\} + \max_{i \in I} |d(x_0, x_i) - d(x_i, x_{i^*})| \right)$$

Problema de la métrica más cercana: Clúster

Análisis de grupos

- Técnica de aprendizaje no supervisada.
- Objetos dentro del mismo clúster son similares.

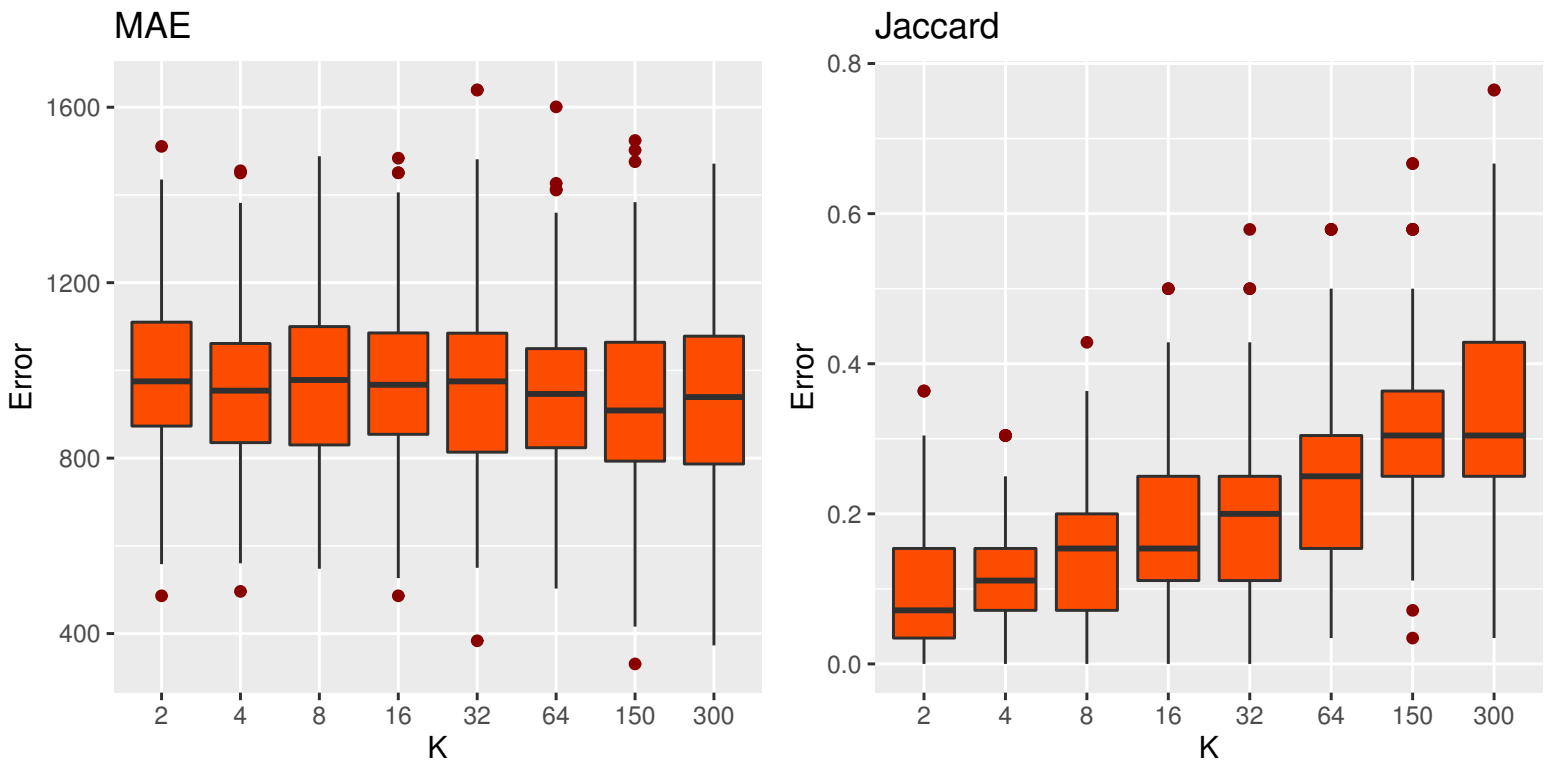
Algoritmos para clúster:

- Agrupamiento jerárquico:
  - Aglomerativo
  - Divisivo.
- Agrupamiento no jerárquico:
  - K-medias.
  - K-medoides:
    - PAM
    - CLARA, CLARANS
    - fastkmed

Selección de K

1. Se generan $n = 3000$ puntos de una distribución normal multivariante con $\mu = \vec{0}_p$ y $\sigma = I_p$, siendo $p = 50$.
2. Se calcula la matriz de distancias, $\mathcal{D}$, usando la distancia euclidiana.
3. Se aplica *fastkmed* a $\mathcal{D}$ para diferentes valores de $K = (2, 4, 8, 16, 32, 64, 150, 300)$ y se obtienen K mediodes $\{C_1, ..., C_K\}$.
4. Se genera un nuevo punto x_0.
5. Se calculan y ordenan las distancias $d(x_0, C_i)$ con $i = 1, ..., K$.
6. En este punto ya hemos calculado K distancias y se calculan las restantes hasta $n(1 - \ell) = 300$ distancias a puntos en los clústeres más cercanos.
7. Para los diferentes valores de K expuestos se calcula el índice de Jaccard y el MAE entre el conjunto real de puntos más cercano y el conjunto de puntos más cercano que se obtiene imputando, para $k = 15$ vecinos.
8. Los pasos $4 - 7$ se repiten $N = 200$ veces.

Selección de K



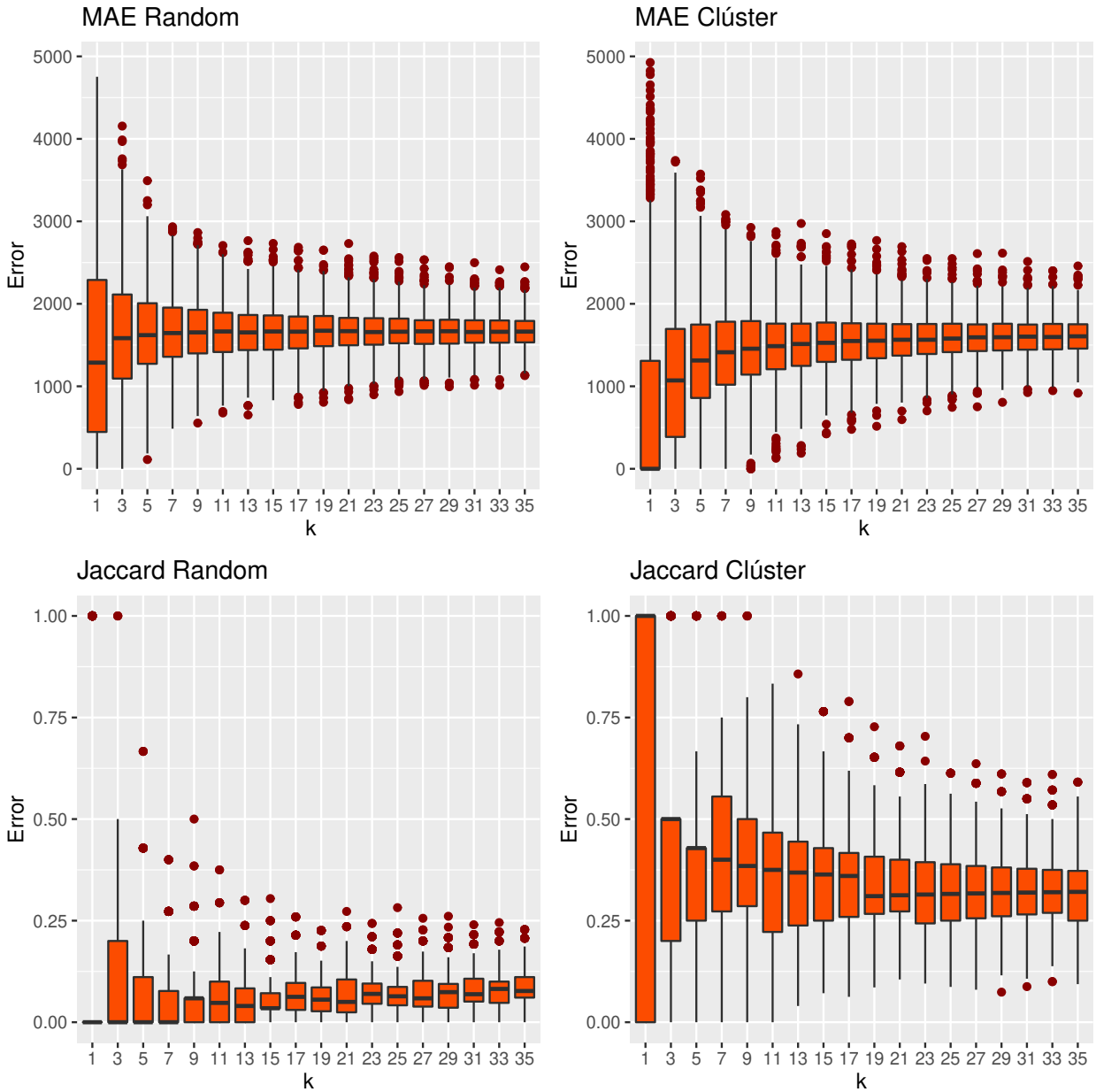
Búsqueda de un valor "óptimo" de clústeres.

$$K = (n - \ell n)/2$$

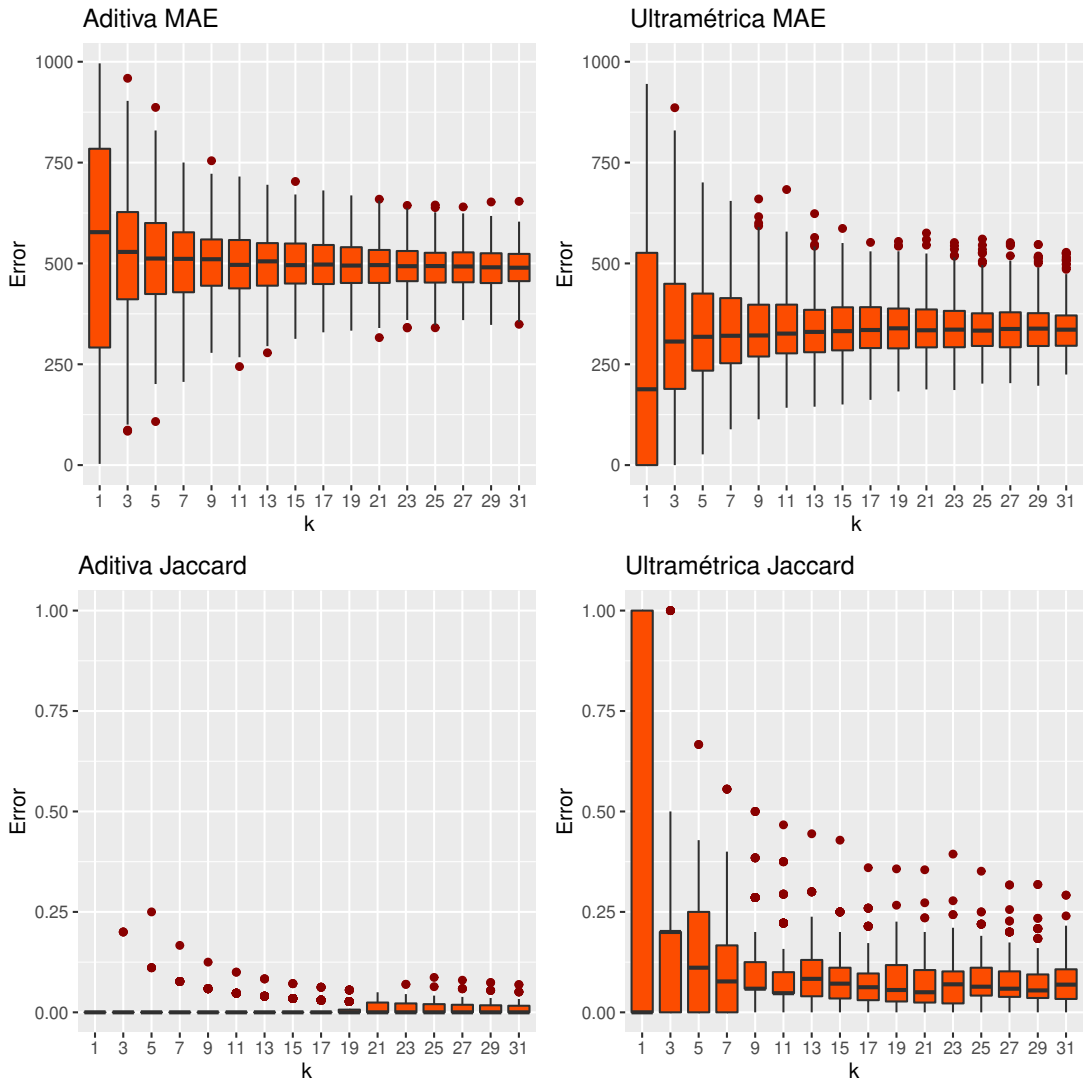
Selección al azar vs. Clúster

1. Se generan $n = 5000$ puntos de una distribución normal multivariante con $\mu = \vec{0}_p$ y $\sigma = I_p$, siendo $p = 50$.
2. Se calcula la matriz de distancias, $\mathcal{D}$, usando la distancia euclidiana.
3. Se genera un nuevo punto x_0.
4. Se calculan las distancias de x_0 a los n puntos.
5. Se calculan las distancias de x_0 a $n(1-\ell)$ puntos al azar y también a la misma cantidad usando $K = (n - ln)/2$ clústeres.
6. Se ordenan dichas distancias y se extrae cuáles son los k puntos más cercanos, el máximo valor que toma k es $\sqrt{n}/2$.
7. Para diferentes valores de k se calcula el índice de Jaccard y el MAE.
8. Los pasos $4-7$ se repiten $N = 1000$ veces.

Selección al azar vs. Clúster

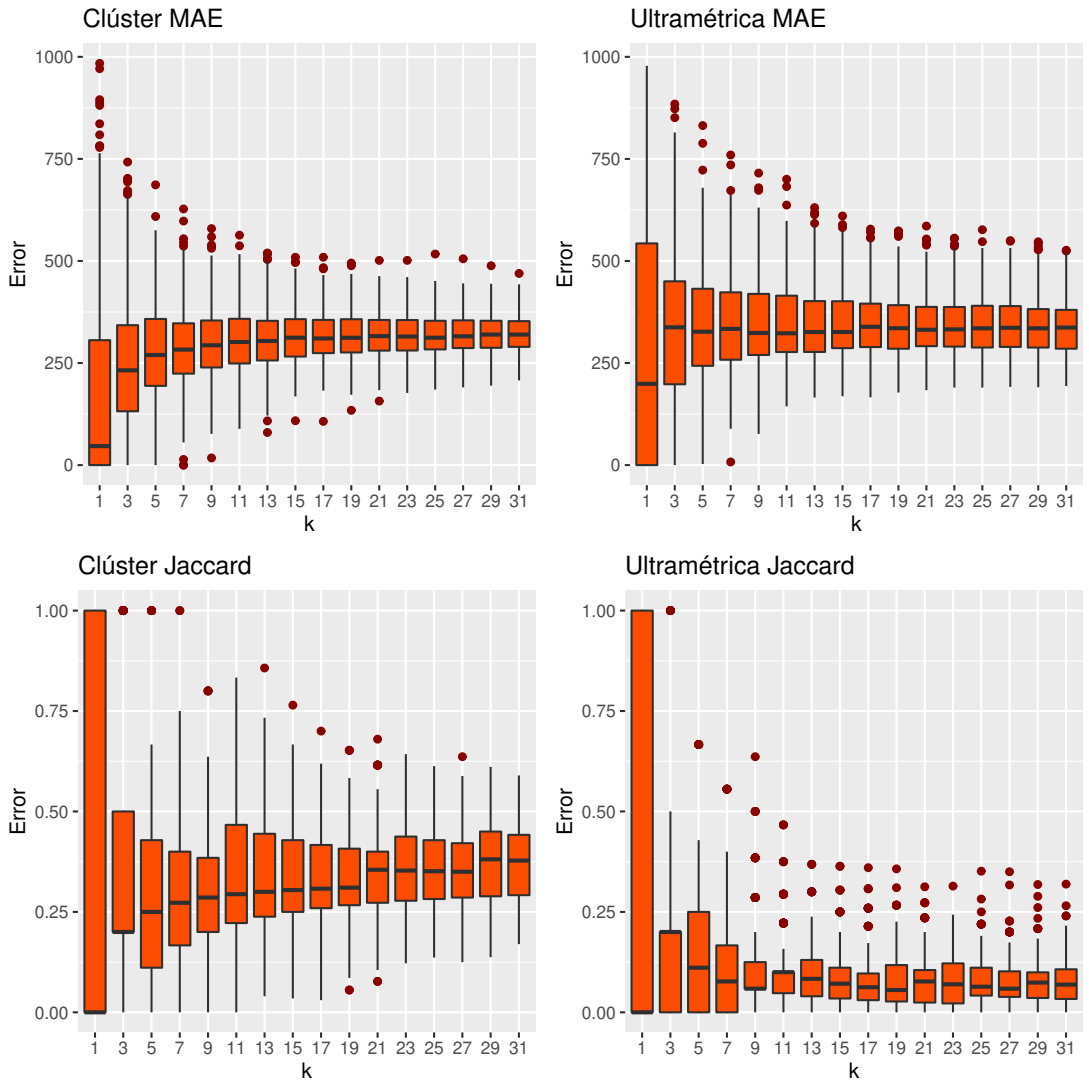


Aditivo vs. Ultramétrica



$n = 1000$, $l = 90\%$, $p = 20$ y $N = 300$.

Ultramétrica vs. Agrupamiento



$n = 1000$, $l = 90\%$, $p = 20$ y $N = 300$.

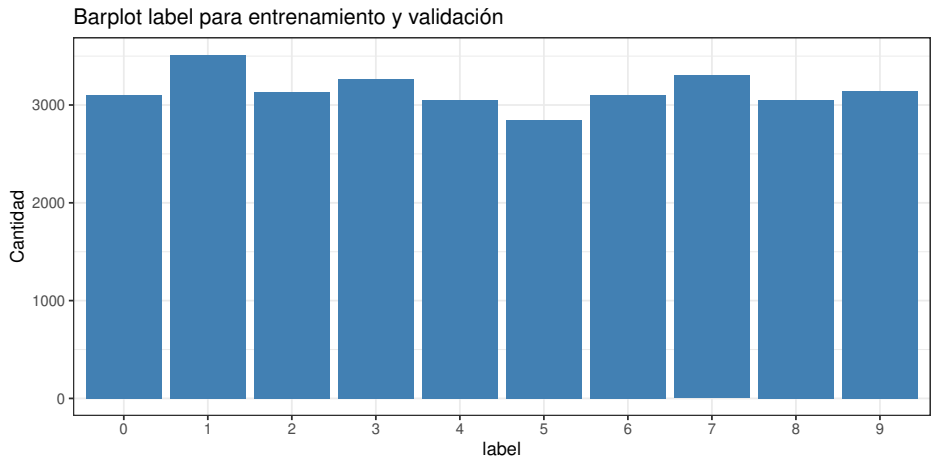
MNIST

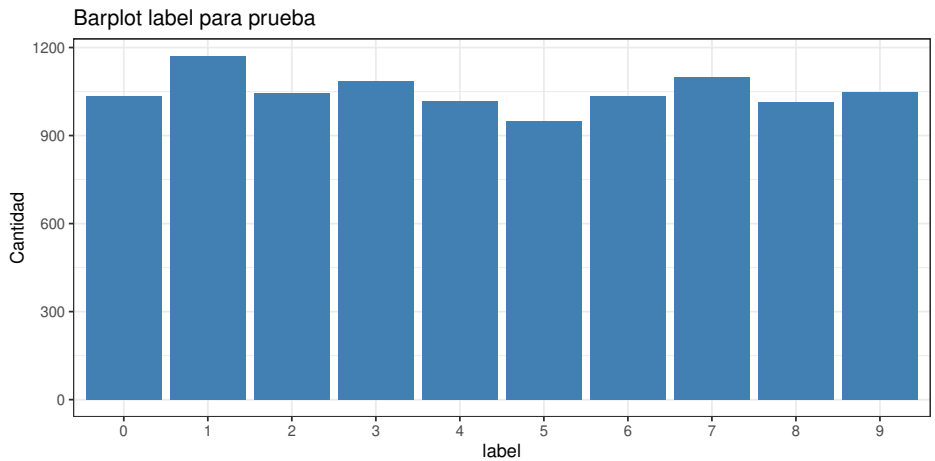
Imágenes en blanco y negro (42000) normalizadas cuyas dimensiones son de 28x28 píxeles en niveles de escala de grises de dígitos escritos a mano.

El problema de clasificación, consiste en, dada una nueva imagen debemos predecir que número tiene escrito.

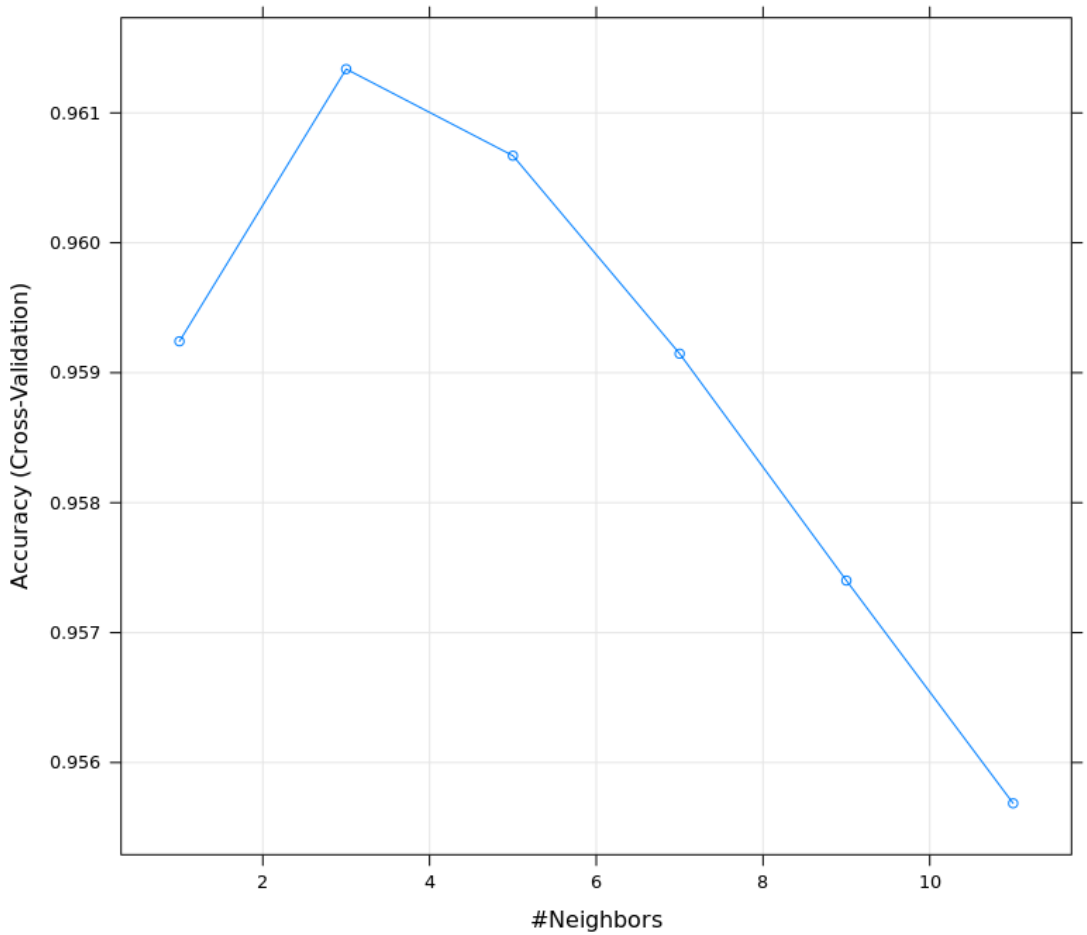
MNIST

El 75% de las imágenes se utiliza para entrenar y el 25% restante de prueba. Esta división se hizo usando un reparto estratificado entre las muestras de entrenamiento y de prueba.





MNIST



Validación cruzada con 5 submuestras, se obtiene una precisión del 96.14%.

MNIST

Matriz de confusión usando k-NN($k = 3$).

	Valor Real									
Predicción	0	1	2	3	4	5	6	7	8	9
0	1024	0	4	2	1	4	6	2	1	6
1	0	1161	8	1	7	1	0	6	10	2
2	2	4	1002	4	0	2	2	3	4	2
3	1	0	5	1043	0	8	1	0	18	10
4	0	1	0	0	985	0	0	1	3	8
5	0	1	1	20	0	920	6	0	16	6
6	5	0	0	1	8	11	1019	0	5	0
7	0	3	21	5	3	0	0	1083	5	10
8	0	1	1	6	1	0	0	0	941	1
9	1	0	2	5	13	2	0	5	12	1002

Precisión: 96.98%

MNIST

Matriz de confusión usando imputación mediante clústeres con $l = 0.75$.

	Valor Real									
Predicción	0	1	2	3	4	5	6	7	8	9
0	1022	0	9	3	2	4	11	0	4	4
1	1	1166	10	4	10	2	1	10	11	1
2	1	1	990	7	0	0	1	6	7	0
3	0	1	2	1030	0	7	0	0	14	3
4	1	2	2	0	971	0	2	6	4	6
5	0	0	0	14	0	920	2	0	19	2
6	6	0	4	3	7	8	1016	0	11	1
7	2	0	21	13	4	2	0	1067	5	12
8	0	0	3	7	0	1	1	0	923	2
9	0	1	3	6	24	4	0	11	17	1016

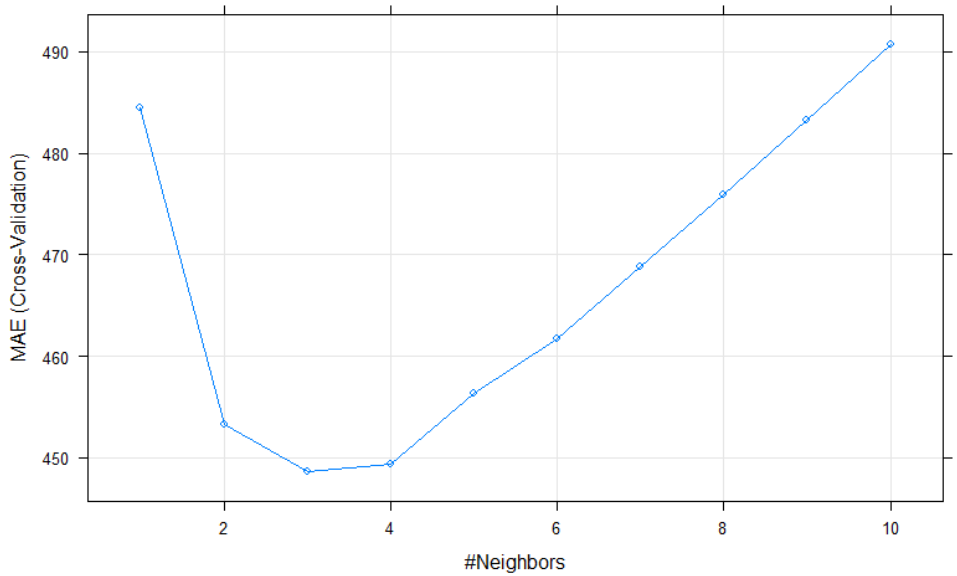
Precisión: 96.42%

DIAMONDS

- Precios y otros diez atributos de casi 54000 diamantes.
- Minimizar el error absoluto medio de la predicción del precio en función de los atributos del diamante.

- price: Precio en USD.
- carat: Peso en quilates del diamante.
- cut: Calidad del corte.
- color: Color del diamante.
- clarity: Claridad del diamante.
- x: Longitud en mm.
- y: Ancho en mm.
- z: Profundidad en mm.
- profundidad: Porcentaje de profundidad total.
- depth: Ancho de la parte superior en relación con el punto más ancho.
- table: Ancho de la parte superior del diamante.

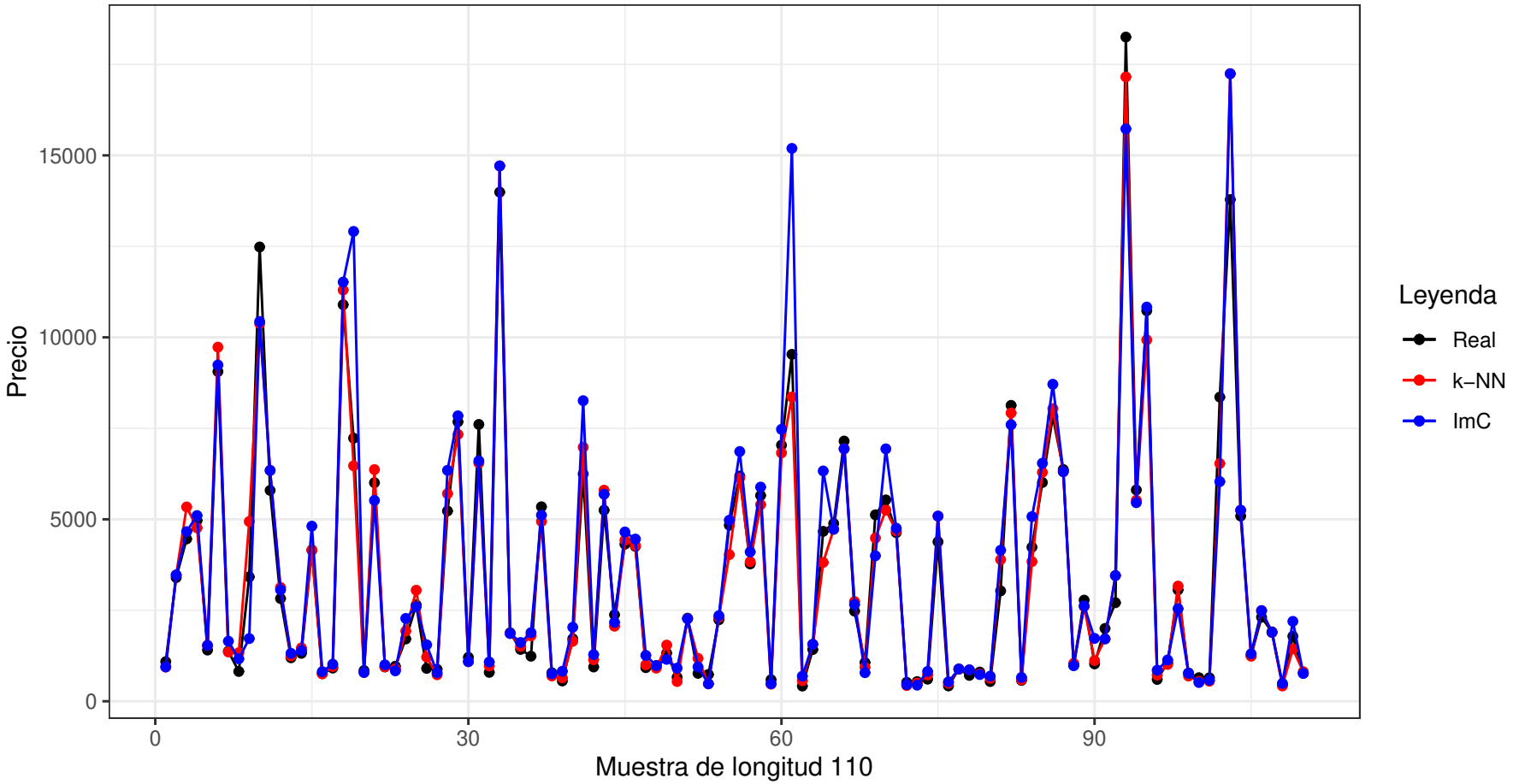
DIAMONDS



- 75% para entrenamiento y 25% para prueba.
- $k = 3$ es el mejor parámetro, $MAE = 448.2$.
- $MAE = 432.881$ usando k-NN en los datos de prueba.
- $MAE = 531.35$ usando el método propuesto con $l = 0.75$.
- $MAE = 639.83$ usando k-NN con muestra de entrenamiento del 25%.

DIAMONDS

Comparación de una muestra entre k-NN, imputación mediante clústeres y valores reales de DIAMONDS



Conclusiones y extensiones

Conclusiones

- Modificaciones al procedimiento k-NN y estudiado el algoritmo triangle fixing con varias opciones de inicialización.
- Observaciones al azar en el conjunto de entrenamiento es inferior a una selección basada en clústeres.
- El procedimiento de imputación basado en clústeres es superior a los algoritmos aditivos y ultramétricos.
- Conjuntos de datos reales que muestran que en problemas de clasificación los resultados son similares al k-NN cosa que no ocurre con el problema de regresión.

Extensiones

- Implementar estos algoritmos en un lenguaje como **C** o **C++**.
- Buscar un número "óptimo" de clústeres, K.
- Selección conjunta del parámetro k del k-NN y del parámetro K del procedimiento de imputación mediante clústeres.

Muchas Gracias