Universidad Nacional Autónoma de México

Facultad de Ciencias, 2024 - 1

Almacenes y Minería de Datos

Tercer Examen Parcial

Integrantes

Adrian Aguilera Moreno	421005200
Israel Hernández Dorantes	318206604

Profesor

Gerardo Avilés Rosas

1. **(10 puntos)** Responde brevemente a las siguientes preguntas:

   - Indica las diferencias entre la limpieza de datos y la transformación de datos.

Limpieza de Datos	**Transformación de Datos**
Identificar y corregir errores, inconsistencias y problemas en los datos crudos.	Modificar y reestructurar datos para adaptarlos a los requisitos específicos del análisis o modelo.
Eliminación de valores atípicos, manejo de valores perdidos, corrección de errores tipográficos, estandarización de formatos, identificación y manejo de duplicados, etc.	Normalización de escalas, codificación de variables categóricas, creación de nuevas características, agregación de datos, cambio de formato de fecha y hora, etc.
Realizada al principio del proceso de preparación de datos para garantizar datos libres de errores.	Suelen ocurrir después de la limpieza y se centran en preparar los datos para análisis, visualización o modelización.
Busca tener datos más precisos y coherentes, eliminando problemas evidentes.	Busca estructurar los datos para que sean más útiles en análisis o modelado, mejorando la eficacia y la interpretabilidad de los resultados.

   - ¿Qué es la propiedad *Apriori*?

     Se refiere a la idea de que si un conjunto de elementos es frecuente, entonces todos sus subconjuntos también deben ser frecuentes. Esta propiedad se basa en la noción de que si un conjunto de artículos es comprado con frecuencia en una transacción, entonces es probable que los subconjuntos más pequeños también se compren con frecuencia.

   - Explica la definición de un centroide en $k-$**means**.

     Un centroide es un punto representativo que se utiliza para representar el "centro" de un grupo de datos. Cada cluster tiene asociado un centroide, que es esencialmente el punto medio o representativo del conjunto de datos que pertenece a ese cluster.

   - Describe el proceso de clasificación.

     (a) **Selección y Preparación de Datos:**
     - Selecciona un conjunto de datos con ejemplos de entrada y etiquetas conocidas.
     - Divide el conjunto en conjuntos de entrenamiento y prueba.

     (b) **Elección del Modelo de Clasificación:**
     - Selecciona un modelo adecuado como máquinas de soporte vectorial, regresión logística, árboles de decisión, etc.

     (c) **Extracción:**
     - Identifica las variables informativas para la tarea de clasificación.

     (d) **Entrenamiento del Modelo:**
     - Entrena el modelo con el conjunto de entrenamiento ajustando sus parámetros internos.

(e) **Validación del Modelo:**

- Evalúa el rendimiento del modelo usando el conjunto de prueba para estimar su capacidad de generalización.

(f) **Ajuste y Optimización del Modelo:**

- Ajusta el modelo según sea necesario, como modificar hiper-parámetros o seleccionar características más relevantes.

(g) **Predicción de Nuevos Datos:**

- Utiliza el modelo entrenado para realizar predicciones en nuevos datos no etiquetados.

(h) **Evaluación del Rendimiento:**

- Evalúa el rendimiento del modelo utilizando métricas como precisión.

2. **(20 puntos)** Supón que un hospital dispone de los datos de **edad** y **grasa corporal** de 18 adultos seleccionados al azar con el siguiente resultado. Justifica tu respuesta:

age	23	23	27	27	39	41	47	49	50
%fat	9.5	26.5	7.8	17.8	31.4	25.9	27.4	27.2	31.2
age	52	54	54	56	57	58	58	60	61
%fat	34.6	42.5	28.8	33.4	30.2	34.1	32.9	41.2	35.7

(a) Calcula la **media**, la **moda**, la **mediana** y la **desviación estándar** de las variables.

i. **Media**

```
> # Mostrar medias para las variables age y fat
> cat("Media para age:", media_age, "\n")
Media para age: 46.44444
> cat("Media para fat:", media_fat, "\n")
Media para fat: 28.78333
```

ii. **Moda**

```
> # Mostrar modas para las variables age y fat
> cat("Moda para age:", moda_age, "\n")
Moda para age: 23 27 54 58
> cat(if(is.null(moda_fat)){
+   "fat es un conjunto amodal."
+   } else{paste("Moda para fat:", moda_fat, "\n")})
fat es un conjunto amodal.
```

iii. **Mediana**

```
> # Mostrar medianas para las variables age y fat
> cat("Mediana para age:", mediana_age, "\n")
Mediana para age: 51
> cat("Mediana para fat:", mediana_fat, "\n")
Mediana para fat: 30.7
```

iv. **Desviación estándar**

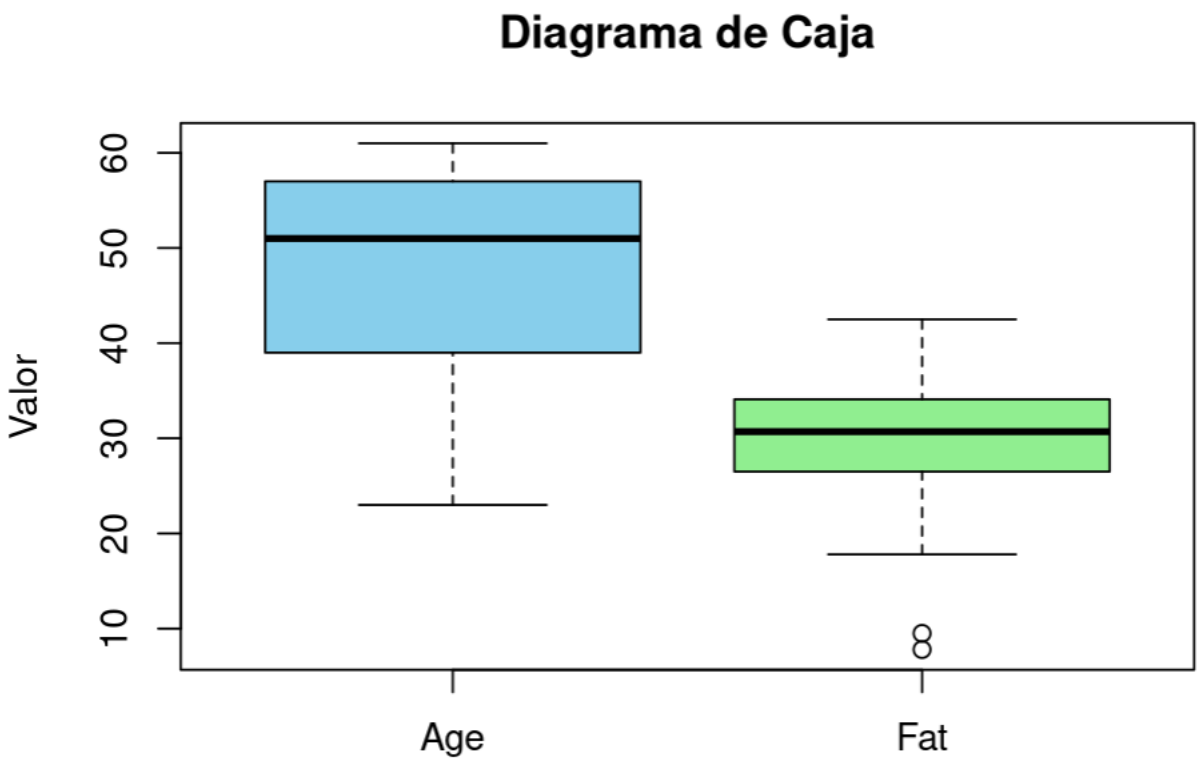
```
> # Mostrar desviación estándar para las variables age y fat
> cat("Desviación estándar para age:", ds_age, "\n")
Desviación estándar para age: 13.21862
> cat("Desviación estándar para fat:", ds_fat, "\n")
Desviación estándar para fat: 9.254395
```

(b) Encuentra (aproximadamente) el **primer cuartil** ($\mathbf{Q}_1$) y el **tercer cuartil** ($\mathbf{Q}_3$) de las variables. Muestra un diagrama de caja de las variables.

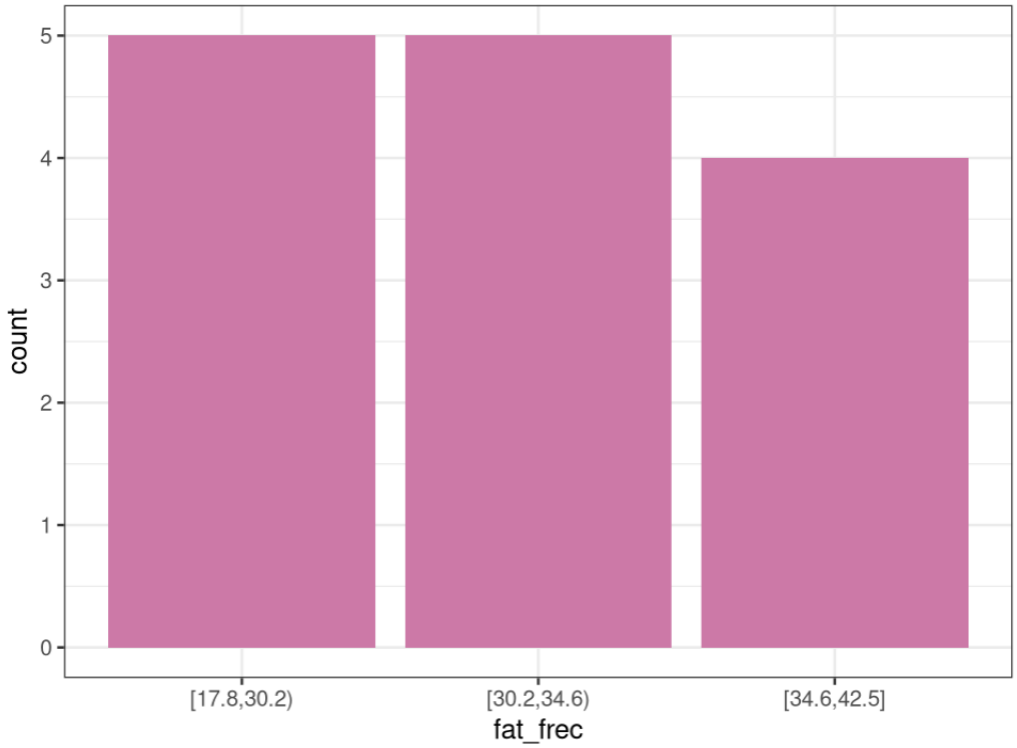
Una aproximación paralos cuartiles 1 y 3 serían

```
> cat("Primer cuartil (Q1) de age:", cuartil_1_age, "\n")
Primer cuartil (Q1) de age: 39.5
> cat("Primer cuartil (Q1) de fat:", cuartil_1_fat, "\n")
Primer cuartil (Q1) de fat: 26.675
> cat("Tercer cuartil (Q3) de age:", cuartil_3_age, "\n")
Tercer cuartil (Q3) de age: 56.75
> cat("Tercer cuartil (Q3) de fat:", cuartil_3_fat, "\n")
Tercer cuartil (Q3) de fat: 33.925
```

El diagrama de caja inducido por nuestro `data` sería



(c) Usa el **suavizado por bin** para suavizar estos datos, usando **bins** de **igual frecuencia** de **tamaño 3**.



(d) Usa la **normalización min-max** para transformar la variable ***age*** en el **rango [0.0,0.1]**.

```
> print(data_normalizado)
 [1] 0.00000000 0.00000000 0.01052632 0.01052632 0.04210526 0.04736842
 [7] 0.06315789 0.06842105 0.07105263 0.07631579 0.08157895 0.08157895
[13] 0.08684211 0.08947368 0.09210526 0.09210526 0.09736842 0.10000000
>
```

(e) Usa la **normalización z-core** para transformar la variable ***% fat***.

```
> print(data_normalizados_z_score)
             [,1]
 [1,] -2.083694688
 [2,] -0.246729622
 [3,] -2.267391195
 [4,] -1.186823509
 [5,]  0.282748545
 [6,] -0.311563683
 [7,] -0.149478530
 [8,] -0.171089884
 [9,]  0.261137191
[10,]  0.628530204
[11,]  1.482178676
[12,]  0.001800946
[13,]  0.498862082
[14,]  0.153080422
[15,]  0.574501820
[16,]  0.444833697
[17,]  1.341704877
[18,]  0.747392650
attr(,"scaled:center")
[1] 28.78333
attr(,"scaled:scale")
[1] 9.254395
>
```

(f) Calcula el **coeficiente de correlación de Pearson**, ¿Están los dos atributos correlacionados positiva o negativamente?

```
> print(matriz_correlacion_parson)
          age       fat
age 1.0000000 0.8176188
fat 0.8176188 1.0000000
>
```

Como podemos notar, la correlación asociada se acerca a 1 en cada celda de la matriz.

De lo anterior, concluimos que la correlación es positiva.

3. **(10 puntos)** Por lo general, estamos muy interesados en las **reglas de asociación** con **alta confianza**, sin embargo, **no siempre serán de interés** las **reglas de asociación** que tengan una **confianza del 100%**, ¿por qué? ¿Por qué considerarías que **reglas de asociación** con un **99% de confianza** pueden **ser interesantes**? ¿qué podrían indicar?

   Las **reglas de asociación** que tienen una confianza del **100%** no siempre pueden ser de interés debido a la **falta de variabilidad** que pueda existir, ya que si una regla de asociación tiene una confianza del 100%, significa que siempre se cumple y no hay ninguna variabilidad en los datos, lo cual no podría aportar información nueva o interesante. También si tenemos una regla de asociación con una confianza del 100% puede ser debido a un **sobreajuste** del modelo a los datos de entrenamiento, lo cual implicaría que el modelo se ha ajustado demasiado a datos específicos utilizados para entrenarlo y podría no generar bien nuevos datos.

   En cambio, las reglas de asociación con un **99%** de confianza resultan ser más interesantes, esto debido a que indica que hay una fuerte relación entre el antecedente y el consecuente en los datos analizados; éstas reglas pueden ayudar a identificar patrones significativos, tendencias o comportamientos que pueden ser útiles en la toma de decisiones. Y por lo tanto, indican **patrones fuertes** y proporcionan **información relevante** que puede ser utilizada para la mejora de procesos y en la **toma de decisiones**. Además este tipo de reglas pueden indicar áreas en las que se pueden realizar **mejoras** o **intervenciones** en los modelos.

4. **(15 puntos)** Considera el conjunto de datos que se muestra en la siguiente tabla:

IDTransacción	ItemsComprados
1	$\{a, b\}$
2	$\{b, c\}$
3	$\{a, b, c\}$
4	$\{d, e, f\}$
5	$\{a, b, c\}$
6	$\{d, f\}$
7	$\{c, d, e, f\}$
8	$\{a, b, c, d, e\}$

a) Aplica el algoritmo ***Apriori*** para la base de datos de transacciones dada, utilizando el soporte mínimo de **0.3** y la confianza mínima de **0.77**. Considera que debes mostrar cómo se realiza el algoritmo, no es suficiente simplemente indicar el resultado final.

Lo primero que se hace es instalar los paquetes que vamos a utilizar:

```
# Instalando paquetes necesarios
install.packages("arules")
install.packages("tidyverse")
```

Después creamos el conjunto de datos para tener la base de datos de transacciones:

```
# Creando la tabla con los datos de transacciones
library(tidyverse)
datos_transacciones <- tibble(
  IDTransaccion = 1:8,
  ItemsComprados = list(c("a", "b"), c("b", "c"), c("a", "b", "c"), c("d", "e", "f"),
                        c("a", "b", "c"), c("d", "f"), c("c", "d", "e", "f"), c("a", "b", "c", "d", "e"))
)
```

Luego, convertimos la columna "ItemsComprados" en un objeto de la clase "transactions":

```
# Convertiendo la columna "ItemsComprados" en un objeto de la clase "transactions"
library(arules)
transacciones <- as(datos_transacciones$ItemsComprados, "transactions")
```

Y ejecutamos el algoritmo ***apriori*** en la base de datos generada, indicando el soporte mínimo de 0.3 en el parámetro de `support` y la confianza mínima de 0.77 en el parámetro de `confidence`:

```
# Ejecutando el algoritmo apriori a la base de datos generada con un soporte minimo de 0.3 y una confianza
# minima de 0.77
rules <- apriori(transacciones, parameter = list(support = 0.3, confidence = 0.77))
```

Y, las reglas de asociación resultantes fueron las siguientes:

```
> # Mostrando las reglas de asociación generadas por el algoritmo
> inspect(rules)
    lhs       rhs support confidence coverage lift count
[1] {f}    => {d} 0.375   1.0        0.375    2.00 3
[2] {e}    => {d} 0.375   1.0        0.375    2.00 3
[3] {a}    => {b} 0.500   1.0        0.500    1.60 4
[4] {b}    => {a} 0.500   0.8        0.625    1.60 4
[5] {b}    => {c} 0.500   0.8        0.625    1.28 4
[6] {c}    => {b} 0.500   0.8        0.625    1.28 4
[7] {a, c} => {b} 0.375   1.0        0.375    1.60 3
>
```

Aquí lo que sucede es que mientras existan *itemsets* frecuentes, de menor a mayor tamaño, se calculará por cada uno de los candidatos su respectivo soporte, construyendo de este modo los conjuntos de *itemsets* frecuentes a partir de los candidatos más frecuentes, y esto se obtiene a partir de aquellos *itemsets* con un soporte mayor o igual al mínimo, 0.3.

b) Suponiendo que tenemos una regla $I_1 \longrightarrow I_2$, describe cómo interpretar la situación cuando la regla tiene: soporte bajo y alta confianza o bien soporte alto y baja confianza.

Cuando una regla $I_1 \longrightarrow I_2$ tiene un **soporte bajo pero una alta confianza**, significa que la regla se aplica a un pequeño número de transacciones, pero cuando ésta se aplica es muy probable que sea correcta; esto puede indicar que los elementos en el lado izquierdo de la regla, i.e. I_1, son raros o poco frecuentes en la base de datos de transacciones, pero cuando están presentes, es muy probable que también estén presentes los elementos en el lado derecho de la regla, i.e. I_2. Entonces, aunque la regla se aplica a un número limitado de transacciones, será muy confiable cuando ésta se aplique. Luego, si esta misma regla tiene un **soporte alto pero una baja confianza**, significa que la regla se aplica a un gran número de transacciones, pero cuando se aplica, no es muy confiable; esto puede indicar que los elementos en el lado izquierdo de la regla, i.e., I_1, son frecuentes en la base de datos, pero no necesariamente están asociados con los elementos en el lado derecho de la regla, i.e. I_2. Entonces, en este caso aunque la regla se aplique a muchas transacciones, ésta no será muy confiable para poder predecir la existencia de los elementos del lado derecho de la regla.

5. **(15 puntos)** Considera los ejemplos de entrenamiento que se muestran en la siguiente tabla para un problema de clasificación binaria:

ID	a_1	a_2	a_3	Clase
1	T	T	1.0	C_1
2	T	T	6.0	C_1
3	T	F	4.0	C_2
4	F	F	7.0	C_1
5	F	T	3.0	C_2
6	F	T	4.0	C_2
7	F	F	8.0	C_2
8	T	F	7.0	C_1
9	F	T	5.0	C_2

a) Calcula la entropía del conjunto de entrenamiento con respecto a la clase C_1?

La entropía del conjunto de entrenamiento con respecto a la clase C_1 es 0.9910761.

```
> # Mostrando resultado
> r_entropia_c1 <- entropia_c1(datos_entrenamiento)
> r_entropia_c1
[1] 0.9910761
```

b) Obtén las ganancias de información de a_1 ya_2.

La ganancia de información de a_1 es: 0.2294368 y la ganancia de información de a_2 es: 0.007214618

```
> # Mostranado ganancia de informacion para a1
> ganancia_a1
[1] 0.2294368
> # Mostranado ganancia de informacion para a2
> ganancia_a2
[1] 0.007214618
```

c) Para a_3, que es un atributo continuo, calcula la ganancia de información para cada división posible.

La ganancia de información de cada división posible es:

i. 0.142690279
ii. 0.002565287
iii. 0.091091008
iv. 0.091091008
v. 0.229436841
vi. 0.072780226
vii. 0.102187171
viii. 0.102187171

```
> # Mostrando las ganancias de información para cada punto medio
> ganancias_informacion_division
[1] 0.142690279 0.002565287 0.091091008 0.091091008 0.229436841 0.072780226 0.102187171 0.102187171
>
```

d) ¿Cuál es la mejor división (entre a_1, a_2 y a_3) según la ganancia de información?

De acuerdo a la ganancia de información de las tres variables:

```
> # Mostrando la ganancia de informacion de a1
> ganancia_a1
[1] 0.2294368
> # Mostrando la ganancia de informacion de a2
> ganancia_a2
[1] 0.007214618
> # Mostrando la ganancia de informacion de a3
> ganancia_a3
[1] 0.9910761
```

Podemos concluir que la división a_3 es la mejor.

6. (20 puntos) Considera los **ejemplos de entrenamiento** que se muestran en la siguiente tabla para un problema de clasificación binaria:

IDCliente	Género	TipoAuto	TallaPlayera	Clase
1	M	Familiar	Ch	C_0
2	M	Deportivo	M	C_0
3	M	Deportivo	M	C_0
4	M	Deportivo	G	C_0
5	M	Deportivo	XG	C_0
6	M	Deportivo	XG	C_0
7	F	Deportivo	Ch	C_0
8	F	Deportivo	Ch	C_0
9	F	Deportivo	M	C_0
10	F	Lujo	G	C_0
11	M	Familiar	G	C_1
12	M	Familiar	XG	C_1
13	M	Familiar	M	C_1
14	M	Lujo	XG	C_1
15	F	Lujo	Ch	C_1
16	F	Lujo	Ch	C_1
17	F	Lujo	M	C_1
18	F	Lujo	M	C_1
19	F	Lujo	M	C_1
20	F	Lujo	G	C_1

(a) **Calcula** el **índice *Gini*** para el **conjunto de entrenamiento**.

```
> cat("El índice Gini del conjunto de entrenamiento es:", indice_gini_total
El índice Gini del conjunto de entrenamiento es: 0.3166667
>
```

(b) **Calcula** el **índice *Gini*** para el **ID Cliente**.

```
> cat("El índice Gini del Género es:", indice_gini_G)
El índice Gini del Género es: 0.5
>
```

(c) **Calcula** el **índice *Gini*** para el **Género**.

```
> cat("El índice Gini del tipo de auto es:", indice_gini_TA)
El índice Gini del tipo de auto es: 0.48
>
```

(d) **Calcula** el **índice *Gini*** para el **Tipo de automóvil**.

```
> cat("El índice Gini de la talla de playeras es:", indice_gini_TP)
El índice Gini de la talla de playeras es: 0.4351852
>
```

(e) **Calcula** el **índice *Gini*** para el **Talla de playera**.

```
> cat("El índice Gini del Clase es:", indice_gini_C)
El índice Gini del Clase es: 0.5
>
```

(f) ¿Qué atributo es mejor para particionar?

R. La talla de playeras, pues es quién tiene el 2do menor índice Ginni.

(g) Explica por qué el ID de cliente no debe usarse como condición de prueba, aunque tenga el ***Ginni*** más bajo.

R. Porque el ID no clasifica algún grupo, sólo enumera. Tomarlo cómo atributo para particionar nos causaria un sobre-ajuste falso.

7. **(10 puntos)** Discute brevemente si cada una de las siguientes o no una tarea de minería de datos.

- Dividir a los clientes de una empresa según su rentabilidad.

  Para este caso, **sí es una tarea de minería de datos**, pues a la división de los clientes implica partir a los clientes en grupos o segmentos más pequeños según ciertas características o comportamientos similares, y como nos interesa dividir a los clientes en función de su **rentabilidad**, la minería de datos puede ayudar a identificar patrones, tendencias y relaciones en los datos que permitan categorizar a los clientes en grupos basados en su contribución a la rentabilidad de algún negocio o empresa.

- Predecir los resultados de lanzar un par de dados.

  Para este caso, **no es una tarea de minería de datos**, esto debido a que predecir los resultados de lanzar un par de dados no requiere un análisis complejo de datos, ni el uso de algoritmos, y la probabilidad de obtener cada resultado en un lanzamiento de dados es conocida y se puede calcular fácilmente. En cambio las tareas de minería de datos se basan en descubrir patrones, relaciones o información valiosa en grandes conjuntos de datos complejos, no en problemas donde las respuestas son completamente predecibles.

- Predecir el precio futuro de las acciones de una empresa utilizando registros históricos.

  Para este caso, **sí es una tarea de minería de datos**, ya que esta tarea implica realizar un análisis de grandes conjuntos de datos para descubrir patrones, tendencias y relaciones ocultas que pueden ayudar a predecir eventos futuros. Por ejemplo, se utilizarían registros históricos de precios de acciones y técnicas de minería de datos para identificar patrones y tendencias en los datos, y éstos se pueden utilizar para construir modelos predictivos que permitan estimar el precio futuro de las acciones de la empresa.

- Monitorear la frecuencia cardíaca de un paciente en busca de anomalías.

  En este caso, **sí es una tarea de minería de datos**, ya que el monitorear la frecuencia cardiaca de un paciente implica recopilar datos sobre su ritmo cardíaco y analizarlos en busca de anomalías o patrones anormales. Por ejemplo, al aplicar técnicas de minería de datos, como algoritmos de detección de anomalías es posible identificar patrones irregulares en los datos de frecuencia cardíaca que podrían indicar un problema de salud del paciente.

- Monitorear y predecir las fallas en una central hidroeléctrica.

  Sí es una tarea de minería de datos, ya que consiste en utilizar *datos históricos* y en tiempo real para poder predecir posibles fallas o problemas en los equipos y tomar medidas preventivas antes de que ocurran, y estos datos se pueden analizar utilizando algoritmos de minería de datos para poder identificar patrones que indiquen posibles fallas o problemas en el funcionamiento de la central.