Tema 2: Técnicas y herramientas de Big Data

Respuesta Múltiple

1. ¿Cuál de las siguientes métricas financieras se utiliza para medir la rentabilidad de una inversión descontando los flujos de caja futuros?
    a) ROI (Retorno de la Inversión)
    b) TIR (Tasa Interna de Retorno)
    c) VAN (Valor Actual Neto)
    d) Ninguna de las anteriores
2. ¿Qué porcentaje aproximado de empresas en España utiliza Big Data según las fuentes?
    a) 50%
    b) 25%
    c) 13.9%
    d) 5%
3. ¿Qué tipo de procesamiento de datos se caracteriza por analizar los datos de manera continua a medida que se generan, en pequeños lotes, y es similar al tiempo real?
    a) Procesamiento por lotes (batch)
    b) Procesamiento en streaming
    c) Procesamiento en tiempo real
    d) Ninguno de los anteriores
4. ¿Cuál de las siguientes etapas no forma parte del modelo de computación MapReduce?
    a) Map
    b) Shuffle
    c) Reduce
    d) Ingestión
5. ¿Cuál es el componente principal de YARN que gestiona los recursos del clúster?
    a) ApplicationMaster
    b) ResourceManager
    c) HDFS
    d) MapReduce
6. ¿Cuál es la estructura de datos fundamental en Apache Spark que permite la tolerancia a fallos y el procesamiento en memoria?
    a) HDFS
    b) DAG
    c) RDD
    d) MapReduce
7. ¿Qué característica clave permite a los RDDs recuperarse de fallos de nodos?
    a) Persistencia en disco
    b) Linaje (parentesco)
    c) Replicación de datos
    d) Procesamiento por lotes
8. ¿Cuál de las siguientes empresas multinacionales utiliza el análisis predictivo para generar recomendaciones de productos a sus clientes?
    a) Apple
    b) Netflix
    c) Amazon
    d) Coca-Cola

9. ¿Qué porcentaje aproximado de empresas en España utilizaban Big Data, según estudios recientes?
    a) 8%
    b) 14%
    c) 25%
    d) 35%
10. ¿Cuál es uno de los principales retos que enfrenta Apache Hadoop en comparación con otras plataformas como Apache Spark?
    a) Escalabilidad limitada.
    b) Falta de tolerancia a fallos.
    c) No permite la recursividad ni la interactividad.
    d) Procesamiento en tiempo real.
11. ¿Qué tipo de procesamiento es más adecuado cuando es necesario analizar datos en pequeños lotes de manera continua a medida que se generan, sin necesidad de resultados inmediatos?
    a) Procesamiento por lotes (Batch Processing)
    b) Procesamiento en streaming
    c) Procesamiento en tiempo real
    d) Procesamiento en MapReduce
12. ¿Cuál es la fórmula del ROI (Retorno de la Inversión)?
    a) ROI = Inversión Inicial / VAN x 100%
    b) ROI = (VAN / Inversión Inicial) x 100%
    c) ROI = VAN - Inversión Inicial
    d) ROI = Inversión Inicial / Beneficios anuales
13. ¿Qué función realiza la etapa Shuffle en el modelo MapReduce?
    a) Acceder a los datos locales de cada máquina y realizar un procesamiento inicial.
    b) Recolectar datos intermedios, almacenarlos temporalmente, y trasladarlos a nuevos nodos.
    c) Procesar los datos intermedios y producir el resultado final.
    d) Asignar tareas a nodos trabajadores.
14. ¿Qué áreas lideran la adopción de Big Data en España?
    a) Agricultura y pesca
    b) Turismo y hostelería
    c) TIC e información y comunicaciones
    d) Comercio y retail
15. ¿Qué significa la sigla RDD en el contexto de Apache Spark?
    a) Recurso de Datos Distribuidos
    b) Red de Datos Dinámica
    c) Conjunto de Datos Distribuido Resiliente
    d) Registro de Datos Digitales
16. ¿Cuál es una de las principales barreras para la implementación de Big Data en las empresas españolas?
    a) Exceso de personal cualificado
    b) Sobredemanda de infraestructura
    c) Falta de disponibilidad de datos
    d) Falta de conocimientos en las empresas

Respuesta Verdadero/Falso

1. El ROI considera el tiempo en el cálculo de la rentabilidad de un proyecto.
2. El procesamiento por lotes se caracteriza por la necesidad de resultados inmediatos.
3. En un sistema Big Data, la capa de visualización es donde se realiza el análisis y procesamiento de los datos.
4. Apache Hadoop es ideal para procesamiento en tiempo real debido a su uso intensivo de la memoria RAM.
5. Spark puede ejecutarse sobre un clúster Hadoop, utilizando su infraestructura de almacenamiento y recursos de cómputo.
6. Las empresas españolas, en su mayoría PYMES, adoptan fácilmente tecnologías BIG data debido a su gran cantidad de recursos.
7. El procesamiento en tiempo real permite cierta pérdida de datos para mejorar la velocidad.
8. La empresa Apple utiliza el análisis de Big Data principalmente para optimizar su cadena de suministro y reducir los costos operativos.
9. El procesamiento por lotes (batch processing) es ideal para aplicaciones que requieren respuestas instantáneas o casi instantáneas a medida que los datos llegan al sistema.
10. La Tasa Interna de Retorno (TIR) es un indicador de rentabilidad que representa el porcentaje de rendimiento que se espera obtener sobre la inversión, y cuanto menor sea la TIR, mejor es la rentabilidad.
11. En el modelo de computación MapReduce, la etapa Shuffle es totalmente transparente para el usuario y se encarga de recopilar los datos intermedios, almacenarlos temporalmente y trasladarlos a los nodos donde se ejecutará la etapa Reduce.
12. Una de las principales barreras para la implementación de Big Data en las empresas españolas es el exceso de personal cualificado disponible en el mercado.

Respuesta Corta

1. Menciona tres empresas internacionales que utilizan Big Data y un ejemplo de cómo lo usan.

2. ¿Cuáles son las principales barreras para la implementación de Big Data en empresas españolas?

3. Describe brevemente las tres etapas del modelo MapReduce.

4. ¿Qué es un RDD en Apache Spark y cuál es su principal ventaja?

5. Explica brevemente la diferencia principal entre el procesamiento por lotes y el procesamiento en streaming en términos de uso de memoria y tiempo.

Problemas

1. Una empresa de marketing digital está considerando invertir en una nueva plataforma de análisis de redes sociales para mejorar sus campañas publicitarias. La inversión inicial incluye la compra del software y la capacitación del personal. Los beneficios se esperan en forma de aumento de ingresos y reducción de costos operativos. Siguiendo los principios de PRINCE2, calcula los flujos de caja netos, el VAN, el ROI y la TIR del proyecto, tanto manualmente como utilizando un programa en Python, con los siguientes supuestos:

- Inversión inicial (Año 0): 80.000 €
- Costos anuales de operación: 8.000 € (años 1 al 4)
- Beneficios netos anuales: 40.000 € (años 1 al 4)
- Horizonte del proyecto: 4 años
- Tasa de descuento: 7% anual

2. Una empresa de logística está evaluando la implementación de un sistema de gestión de almacén (SGA) para mejorar la eficiencia y reducir los errores en el manejo de inventario. El proyecto requiere una inversión inicial en software y hardware, con flujos de caja netos que varían según los ahorros y la mejora en la eficiencia. Calcula el VAN, ROI y TIR usando los siguientes datos:

- Inversión inicial (Año 0): 150.000 €
- Costos y Beneficios:
  - Año 1: Costo 35.000€, Beneficio 70.000€
  - Año 2: Costo 35.000€, Beneficio 75.000€
  - Año 3: Costo 35.000€, Beneficio 80.000€
  - Año 4: Costo 35.000€, Beneficio 90.000€
  - Año 5: Costo 35.000€, Beneficio 75.000€
- Horizonte del proyecto: 5 años
- Tasa de descuento: 8.5% anual

3. Una empresa de energías renovables está considerando un proyecto para instalar paneles solares en un edificio comercial. Ya se ha realizado un análisis y se sabe que el VAN del proyecto es positivo, pero se necesita calcular la inversión inicial necesaria para obtener ese VAN específico. Se conocen los siguientes datos:

- VAN: 60.000 €
- Flujos de caja netos esperados:
  - Año 1: 45.000 €
  - Año 2: 55.000 €
  - Año 3: 65.000 €
- Tasa de descuento: 9%

4. Una empresa de comercio electrónico quiere analizar las valoraciones de productos por parte de sus clientes. Los datos se presentan en formato: ["ID_Producto, Valoración"]. El objetivo es calcular la valoración media para cada producto. La empresa cuenta con 3 máquinas para el procesamiento distribuido.
Los datos son: ["P101, 4", "P102, 5", "P101, 3", "P103, 4", "P102, 4", "P101, 5", "P103, 5", "P102, 3"]

a) División de datos y asignación a máquinas: Divide los datos en fragmentos y simula la asignación a las 3 máquinas disponibles.
b) Etapas de MapReduce: Describe lo que sucede en cada etapa (Mapping, Shuffle, Reduce) para obtener el resultado final.
c) Implementación en Python: Implementa el ejercicio en Python, dividiendo los datos, ejecutando las etapas de Map, Shuffle y Reduce, y mostrando los resultados finales.

5. Una compañía de transporte necesita analizar los tiempos de viaje entre diferentes ciudades para optimizar sus rutas. Los datos se proporcionan como: ["CiudadOrigen-CiudadDestino, TiempoViajeEnHoras"]. El objetivo es determinar el tiempo de viaje promedio entre cada par de ciudades. Se cuenta con 4 máquinas para el procesamiento.
Los datos son: ["Madrid-Barcelona, 6", "Madrid-Valencia, 3", "Barcelona-Valencia, 4", "Madrid-Barcelona, 7", "Valencia-Madrid, 3", "Barcelona-Madrid, 6", "Valencia-Barcelona, 5", "Madrid-Valencia, 4"]

a) División de datos y asignación a máquinas: Divide los datos en fragmentos y asigna cada fragmento a una de las 4 máquinas.
b) Etapas de MapReduce: Describe qué sucede en cada etapa (Mapping, Shuffle, Reduce).
c) Implementación en Python: Implementa la solución en Python, simulando el proceso MapReduce.

6. Una plataforma de streaming quiere analizar el número de veces que los usuarios ven diferentes películas. Los datos se registran como ["ID_Usuario, ID_Pelicula"]. El objetivo es determinar cuántas veces se ha visto cada película. La empresa cuenta con 2 máquinas para el procesamiento distribuido.
Los datos son: ["U101, M201", "U102, M202", "U101, M201", "U103, M203", "U102, M201", "U101, M202", "U103, M202", "U102, M202"]

a) División de datos y asignación a máquinas: Divide los datos y asigna cada fragmento a una de las 2 máquinas.
b) Etapas de MapReduce: Describe qué ocurre en cada etapa (Mapping, Shuffle, Reduce).
c) Implementación en Python: Implementa el ejercicio en Python, simulando el proceso de MapReduce.

Soluciones

Respuesta Múltiple

1. c, 2. c, 3. b, 4. d, 5. b, 6. c, 7. b, 8. c, 9. b, 10. c, 11. b, 12. b, 13. b, 14. c, 15. c, 16. d

Respuesta Verdadero/Falso

1. **Falso**. El ROI no considera el tiempo.
2. **Falso**. El procesamiento por lotes se utiliza cuando no se necesitan resultados inmediatos.
3. **Falso**. La capa de análisis es donde se procesan los datos, mientras que la visualización es donde se presentan los resultados.
4. **Falso**. Hadoop está diseñado para procesamiento por lotes y utiliza el disco duro. Spark es el que usa la memoria RAM.
5. **Verdadero**.
6. **Falso**. Las PYMES tienen dificultades para adoptar tecnologías BIG data.
7. **Falso**. No permite la pérdida de datos.
8. **Falso**. Apple utiliza el análisis de Big Data para personalizar la experiencia del usuario y mejorar la eficiencia operativa
9. **Falso**. El procesamiento en streaming o tiempo real es el más adecuado para respuestas casi instantáneas, mientras que el procesamiento por lotes trabaja con grandes volúmenes de datos que se procesan sin urgencia.
10. **Falso**. Cuanto mayor sea la TIR, mejor es la rentabilidad del proyecto.
11. **Verdadero**.
12. **Falso**. La falta de personal cualificado es una de las principales barreras.

Respuesta Corta

1.
   - **Amazon**: utiliza análisis predictivo para sistemas de recomendación.
   - **Netflix**: analiza el comportamiento de sus usuarios para la producción de contenido.
   - **Coca-Cola**: utiliza información de consumidores para el desarrollo de productos y análisis de redes sociales.
2.
   - Falta de personal cualificado
   - Falta de infraestructura
   - Disponibilidad de datos
   - Privacidad y seguridad
   - Falta de conocimientos
3.
   - **Map**: Cada nodo procesa datos locales y genera datos intermedios clave-valor.
   - **Shuffle**: Los datos intermedios se reorganizan y reordenan.
   - **Reduce**: Se procesan los datos intermedios para producir el resultado final.

4. Un **RDD** es un Conjunto de Datos Distribuido Resiliente que permite el procesamiento en memoria y la tolerancia a fallos. Su principal ventaja es la capacidad de recuperarse de fallos gracias a su "linaje".

5. El procesamiento por lotes utiliza el disco duro y procesa grandes volúmenes de datos en una sola ejecución, sin necesidad de resultados inmediatos. El procesamiento en streaming utiliza la memoria RAM, procesa datos en pequeños lotes de manera continua y secuencial a medida que llegan, y se usa para obtener resultados casi instantáneos.

Problemas

1. Solución Problema 1
    1. Flujos de caja netos:
        - Año 0: -80.000 €
        - Año 1: 40.000 € - 8.000 € = 32.000 €
        - Año 2: 40.000 € - 8.000 € = 32.000 €
        - Año 3: 40.000 € - 8.000 € = 32.000 €
        - Año 4: 40.000 € - 8.000 € = 32.000 €
    2. VAN: Utilizando la fórmula del VAN: VAN = -80000 + 32000/(1.07) + 32000/$(1.07)^2$ + 32000/$(1.07)^3$ + 32000/$(1.07)^4$ VAN ≈ -80000 + 29906.54 + 27949.05 + 26120.61 + 24411.78 ≈ 28387.98 €
    3. ROI: Utilizando la fórmula del ROI: ROI = 28387.98 / 80000 ≈ 0.3548 = 35.48%
    4. TIR: La TIR se calcula igualando el VAN a cero. Para este problema, la TIR se calcula con métodos numéricos o software financiero. Usando un programa en Python como el del ejemplo, se obtiene una TIR aproximada de 19.29%

2. Solución Problema 2
    1. Flujos de caja netos:
        - Año 0: -150.000 €
        - Año 1: 70.000 € - 35.000 € = 35.000 €
        - Año 2: 75.000 € - 35.000 € = 40.000 €
        - Año 3: 80.000 € - 35.000 € = 45.000 €
        - Año 4: 90.000 € - 35.000 € = 55.000 €
        - Año 5: 75.000 € - 35.000 € = 40.000 €
    2. VAN: Aplicando la fórmula del VAN: VAN = -150000 + 35000/(1.085) + 40000/$(1.085)^2$ + 45000/$(1.085)^3$ + 55000/$(1.085)^4$ + 40000/$(1.085)^5$ VAN ≈ -150000 + 32258.06 + 34008.16 + 35441.83 + 39789.92 + 26546.31 ≈ 18044.28 €
    3. ROI: Usando la fórmula del ROI: ROI = 18044.28 / 150000 ≈ 0.1203 = 12.03%
    4. TIR: La TIR se calcula igualando el VAN a cero. Utilizando métodos numéricos o software financiero, la TIR aproximada es 11.19%

3. Solución Problema 3
    1. Cálculo de la inversión inicial:
        - La fórmula del VAN es: VAN = -I + F1/(1+k) + F2/$(1+k)^2$ + F3/$(1+k)^3$
        - Despejamos la inversión inicial (I): I = F1/(1+k) + F2/$(1+k)^2$ + F3/$(1+k)^3$ - VAN
        - Sustituyendo los valores conocidos: I = 45000/(1.09) + 55000/$(1.09)^2$ + 65000/$(1.09)^3$ - 60000
        - I ≈ 41284.40 + 46359.63 + 50111.04 - 60000 I ≈ 77755.07 € *La inversión inicial necesaria para obtener un VAN de 60.000€ es de aproximadamente 77755.07 €.

4. Solución Ejercicio 1
    a) División de datos y asignación a máquinas:
        - Máquina 1: ["P101, 4", "P102, 5", "P101, 3"]
        - Máquina 2: ["P103, 4", "P102, 4", "P101, 5"]
        - Máquina 3: ["P103, 5", "P102, 3"]
    b) Etapas de MapReduce:
        - Mapping: Cada máquina procesa sus fragmentos de datos. La función Map transforma cada registro en un par clave-valor: la clave es el ID del producto y el valor es la valoración. Por ejemplo, ["P101, 4"] se convierte en ("P101", 4).
        - Shuffle: Los resultados de la etapa Map se agrupan por clave (ID del producto). Todos los valores correspondientes a un mismo ID de producto se envían a la misma máquina para la etapa Reduce.

- Reduce: Cada máquina calcula la media de las valoraciones para cada ID de producto. Por ejemplo, para "P101", se reciben los valores y se calcula la media: (4 + 3 + 5) / 3 = 4.

c) Implementación en Python:

```
def map_function(record):
    product, rating = record.split(", ")
    return product, int(rating)

def reduce_function(key, values):
    average_rating = sum(values) / len(values)
    return key, average_rating

data = ["P101, 4", "P102, 5", "P101, 3", "P103, 4", "P102, 4", "P101, 5", "P103, 5", "P102, 3"]

# Simulación de la etapa de Map
mapped_data = [map_function(record) for record in data]

# Simulación de la etapa de Shuffle
shuffled_data = {}
for key, value in mapped_data:
    if key not in shuffled_data:
        shuffled_data[key] = []
    shuffled_data[key].append(value)

# Simulación de la etapa de Reduce
reduced_data = {key: reduce_function(key, values) for key, values in shuffled_data.items()}
print(reduced_data)
```

5. Solución Ejercicio 2

a) División de datos y asignación a máquinas:
   - Máquina 1: ["Madrid-Barcelona, 6", "Madrid-Valencia, 3"]
   - Máquina 2: ["Barcelona-Valencia, 4", "Madrid-Barcelona, 7"]
   - Máquina 3: ["Valencia-Madrid, 3", "Barcelona-Madrid, 6"]
   - Máquina 4: ["Valencia-Barcelona, 5", "Madrid-Valencia, 4"]

b) Etapas de MapReduce:
   - Mapping: Cada máquina procesa su fragmento. La función Map emite pares clave-valor donde la clave es el par de ciudades (por ejemplo, "Madrid-Barcelona") y el valor es el tiempo de viaje.
   - Shuffle: Los pares clave-valor se agrupan por clave (par de ciudades). Todos los tiempos de viaje para un mismo par de ciudades se envían a la misma máquina para la etapa Reduce.
   - Reduce: Cada máquina calcula el promedio del tiempo de viaje para cada par de ciudades. Por ejemplo, para Madrid-Barcelona, se reciben los valores `` y se calcula la media: (6 + 7) / 2 = 6.5.

c) c) Implementación en Python:

```
def map_function(record):
    cities, time = record.split(", ")
    return cities, int(time)

def reduce_function(key, values):
    average_time = sum(values) / len(values)
    return key, average_time

data = ["Madrid-Barcelona, 6", "Madrid-Valencia, 3", "Barcelona-Valencia, 4", "Madrid-Barcelona, 7",
"Valencia-Madrid, 3", "Barcelona-Madrid, 6", "Valencia-Barcelona, 5", "Madrid-Valencia, 4"]

# Simulación de la etapa de Map
mapped_data = [map_function(record) for record in data]

# Simulación de la etapa de Shuffle
shuffled_data = {}
for key, value in mapped_data:
    if key not in shuffled_data:
        shuffled_data[key] = []
    shuffled_data[key].append(value)

# Simulación de la etapa de Reduce
reduced_data = {key: reduce_function(key, values) for key, values in shuffled_data.items()}
print(reduced_data)
```

6. Solución Ejercicio 3

a) División de datos y asignación a máquinas:
   - Máquina 1: ["U101, M201", "U102, M202", "U101, M201", "U103, M203"]
   - Máquina 2: ["U102, M201", "U101, M202", "U103, M202", "U102, M202"]

b) Etapas de MapReduce:
   - Mapping: Cada máquina procesa sus datos. La función Map transforma cada registro en un par clave-valor. La clave es el ID de la película y el valor es 1 (indicando una visualización). Por ejemplo, ["U101, M201"] se transforma en ("M201", 1).
   - Shuffle: Los pares clave-valor se agrupan por clave (ID de película). Todas las visualizaciones para una misma película se envían a la misma máquina para la etapa Reduce.
   - Reduce: Cada máquina cuenta el número total de visualizaciones para cada película. Por ejemplo, para M201, se reciben los valores y se calcula la suma: 1 + 1+ 1 = 3.

c) Implementación en Python:

```
def map_function(record):
    user, movie = record.split(", ")
    return movie, 1

def reduce_function(key, values):
    total_views = sum(values)
    return key, total_views

data = ["U101, M201", "U102, M202", "U101, M201", "U103, M203", "U102, M201", "U101, M202", "U103,
M202", "U102, M202"]

# Simulación de la etapa de Map
mapped_data = [map_function(record) for record in data]

# Simulación de la etapa de Shuffle
shuffled_data = {}
for key, value in mapped_data:
    if key not in shuffled_data:
        shuffled_data[key] = []
    shuffled_data[key].append(value)

# Simulación de la etapa de Reduce
reduced_data = {key: reduce_function(key, values) for key, values in shuffled_data.items()}
print(reduced_data)
```