Organización de Datos 75.06. Primer Cuatrimestre de 2018. Examen parcial, segunda oportunidad: Resolucion

1- Map innecesarios descuento de 3 puntos. Join sin claves compatibles descuento de 5 puntos. No filtrar lo antes posible descuento de al menos 3 puntos.

2- a) Se comienzan filtrando los productos de la categoría "Men" para luego joinear contra las ventas. Una vez hecho eso, hay que agrupar por mes y año para aplicar un transform y obtener el promedio de ventas (será algo del estilo transform(lambda x: x.mean()). Al resultado del transform podemos agregarlo como una columna nueva del dataframe y luego simplemente verificamos que el importe sea mayor que el promedio mensual y sobre eso se podrán obtener los títulos de los productos.. Si lo resuelven sin transform -5.
b) Para empezar, es necesario calcular el total de ventas por dia por producto (o quedarnos con 1 venta por dia por producto, lo que nos interesa es saber si tuvieron alguna venta cada dia). Luego, para cada producto y utilizando shift() podemos obtener el valor de la fecha siguiente, y así comparar la diferencia de fecha. Una vez hecho esto, podemos generar una nueva columna que tenga 0 si la diferencia de fecha es 1 (y se trata de días consecutivos), o 1, si la diferencia es mayor que uno (se trata de días no consecutivos). Al aplicar cumsum a esta nueva columna vamos a poder agrupar por bloque de días consecutivos, y haciendo un size sabremos de cuántos días consecutivos se trata. Tras hacer un reset index podemos quedarnos con los productos que se vendieron más días consecutivos que es lo pedido.

3- V/F
a) Verdadera. La justificación sería la siguiente: Suponiendo un archivo de tamaño N (en bytes) que lo comprimimos en 1 byte entonces obtenemos un archivo comprimido de tamaño N-1. Dado que hay 256^N archivos posibles, y hay 256^(N-1) archivos de N-1 bytes entonces la proporción 256^(N-1) / 256^(N) = 1 / 256 < 1%.
b) Falso. Supongamos archivos de tamaño N. Dado que hay 256^N archivos posibles y 256^(N/2) archivos de la mitad de tamaño posibles, si encuentro ese compresor que me piden entonces tiene que ser cierto que 256^(N/2) / 256^(N) = 1 / 256^(N/2) sea mayor o igual a 0.5. Se puede verificar que el valor máximo que toma esta función es cuando N = 2 y dicho valor es 1/256 (Asumiendo N>1) que es menor que 0.5%. Esto quiere decir que si pudiera encontrar dicho compresor entonces solo en el mejor de los casos (cuando N=2) podría comprimir a la mitad de tamaño solo el 1/256 de los archivos posibles para ese N=2.
c) Verdadera. El número de strings de longitud n es 2^n . Por otro lado, la cantidad de descripciones de longitud < n is $2^0 + 2^1 + 2^2 + ... + 2^{n-1} = 2^n - 1$, lo que es menor a 2^n . Por lo tanto, existe al menos un string de longitud n que sea incompresible.
d) La respuesta es Falsa. Si bien es cierto que K(XY) = K(YX), no necesariamente va a ocurrir que para un algoritmo dado el nivel de compresión se mantenga invirtiendo el orden de los archivos. Debe indicar esto y proponer un contraejemplo.

4- Cada error de cuenta descuentra 3 puntos (son cuentas muy sencillas). Cada item (a,b y c) valen 5 puntos cada uno.

Lo que se está pidiendo es simular 3 permutaciones al azar de la matriz. El algoritmo se encuentra explicado en el apunte de la materia.
Para estimar la semejanza de Jaccard, hay que simplemente hacer la división entre la cantidad de minhash que coinciden y dividirla por 3. Es decir que si los 3 minhash coinciden entonces la semejanza es 1. Para saber si la estimación coincide, se basta con calcular la semejanza de jaccard de los documentos mencionados y ver si los valores coinciden
Un par (Si,Sj) es candidato a ser similar si sus 3 valores de minhash son iguales. Hay que encontrar si hay pares que cumplan esa condición.

5- Deben partir armando el índice invertido con FC parcial (n=3). Si arman mal la estructura del índice, no concatenan los términos y los punteros, el ejercicio vale cero. Luego, sobre la estructura armada deben explicar cómo se realiza una búsqueda binaria sobre el índice, de cada uno de los dos términos. Si hacen busqueda secuencial vale cero ya que carece de sentido tener un índice. Deben contar los accesos al índice, y luego los accesos al léxico para reconstruir cada término, y por último una vez encontrado cada uno de los dos términos buscados acceder a los punteros a documentos.

6- No es necesario reconstruir la matriz original para obtener las respuestas y los errores de cuenta descuentan 3 puntos:

1. La matriz de autovectores nos permite encontrar un espacio sobre el cual proyectar nuestros datos de forma tal de conservar la variabilidad de los mismos. Se puede interpretar también que los autovectores intentan encontrar "conceptos" en nuestros datos. En este ejercicio, cada autovector representa conceptos de categorías de películas: la primer categoría reúne a Matrix, Alien y Starwars (podría ser, por ejemplo, la categoría ciencia ficción) mientras que la segunda categoría reúne a Casablanca y Titanic (podría ser Romance por ejemplo).
2. Joe y Jack solo están calificando películas de la primer categoría (ciencia ficción), por eso los ceros en la segunda columna. Dado que Jack tiene .70 en la columna U, podemos pensar que Jack dió calificaciones más altas que Joe en las películas de dicha categoría. (Es decir, a Jack le gustan más las películas de ciencia ficción).
3. No. La respuesta está en el enunciado. La matriz es de rango dos con lo cual podemos reconstruir la matriz original usando esta descomposición.

7- Se evaluará la calidad de la solución propuesta. Si faltan referencias o títulos se descuentan de 3 a 5 ptos por cada cosa. Si no se explica que quieren comunicar o por que se tomaron las decisiones de diseño tomadas se descuentan 5ptos.