Organización de Datos 75.06. Segundo Cuatrimestre de 2017. Examen parcial, primera oportunidad: CRITERIO

1) Si en algún lado hacen un groupByKey se hace un descuento de 10 puntos. En ningún caso son necesarios los datos de cada registro. Si filtran después de hacer el resto de las operaciones, descuento de 2 puntos (es mucho más eficiente filtrar antes y solo trabajar con un conjunto acotado, podríamos tener todos los patentamientos de la historia).

2) Si hacen mal el join/merge se descuentan 7ptos, si hay problemas en como agrupan o en el manejo dentro del grupo se descuentan 7 ptos. Si agrupan por playlist y obtienen el cantante que mas aparece con un value_counts está bien también.

3) V/F

a) Todo archivo con complejidad de kolmogorov baja tendrá una entropía de Shannon baja.

(FALSO). Podemos utilizar el ejemplo del numero aureo. Se puede calcular simplemente como (1 + Raiz(5)) / 2. Su complejidad de kolmokorov es baja. Pero si vemos el numero: 1,618033988749... nos dará un entropía alta.

b) Es imposible que un archivo comprimido mediante huffman estático de orden 1 iguale la máxima compresión dada por la entropía del mismo.

(FALSO). Hay varias formas de demostrarlo. Por ejemplo si el archivo contiene todos los caracteres como equiprobables. la entropía indicara 8bits por byte y la codificación por huffman estática será el mismo resultado. Otra manera es me quedan cada una de las frecuencias potencias de 2.

c) Podemos determinar la longitud final del archivo comprimido utilizando huffman estático de orden 1, calculando la entropía y multiplicándola por la cantidad de caracteres del archivo

(VERDADERO). La entropía se calcula como la sumatorio de la probabilidad de cada caracter por la longitud de su código.

LongFile * Sumatoria (probCharX * LongCodX) = Sumatoria (LongFile * probCharX * LongCodX)

y LongFile * probCharX = CantidadCharXEnFile

entonces: Sumatoria (CantidadCharXEnFile * LongCodX) = longitud del file comprimido.

Si indican que es FALSO y justifican que por ser estático deben guardar también la tabla de frecuencias (y no saben su tamaño) o que la entropía podría dar un número no entero y huffman genera códigos de una longitud entera de bits eta bien también.

d) Tenemos 2 archivos, uno con longitud pequeña y el otro muy grande que se comprimen utilizando huffman estático de orden 5. Si observamos que tienen la mismas tablas de frecuencias podemos afirmar que el cociente entre el tamaño del archivo sin comprimir y el tamaño del archivo comprimido será similar.

(FALSO): Se tiene que tener en cuenta el tamaño que implica almacenar la tablas de frecuencias. Al tener los dos archivos mismas distribución de probabilidades. La tablas de frecuencias ocuparan el mismo tamaño. Al ser el orden del archivo grande entonces quedaran muchos contextos a almacenar. Por lo tanto si dividimos el tamaño de cada uno de los archivos comprimidos por su tamaño original el cociente favorecerá al archivo grande.

4) Si utilizan tf diferente a bm25: -5pts
- Si hacen cálculos de tf-idf para otros términos o documentos (error conceptual): -3pt
- Errores de cálculo: -3pts
- Si agregan normalización de longitud de documentos: -6pts

5) Si solo descomprimen hasta el caso especial vale un par de puntos nomás.
Si no tienen en cuenta para el tamaño del comprimido que los caracteres son de 9 bits descuento de 3 puntos.

6) La probabilidad es (1-d/180) siendo d la distancia (ángulo) de allí dada la probabilidad pueden despejar el ángulo que debería ser 9 grados. Luego deberían dar dos vectores que estén a 9 grados entre sí y un hiperplano que los bisecte.

7) Suponiendo una distribución uniforme a cada posición le tocan 10 claves es decir que hacen falta 100 lugares (n al cuadrado) es decir que necesitamos 100 * 1000 = 100.000 lugares mas la tabla original es decir 101.000 lugares de 32 bytes cada uno.