114

+30

A

Organización de Datos 75.06. Segundo Cuatrimestre de 2018. Examen parcial, primera oportunidad

Importante: Antes de empezar complete nombre y padrón en el recuadro. Lea bien todo el enunciado antes de empezar. Para aprobar se requiere un mínimo de 60 puntos (60 puntos = 4) con al menos 20 puntos entre los ejercicios 1 y 2. Este enunciado debe ser entregado junto con el parcial si quiere una copia del mismo puede bajarla del grupo de la materia. En el ejercicio 3 elija 2 de los 4 ejercicios y resuelva única y exclusivamente 2 ejercicios. Si tiene dudas o consulta levante la mano, está prohibido hablar desde el lugar, fumar o cualquier actividad que pueda molestar a los demás. El criterio de corrección de este examen está disponible en forma pública en el grupo de la materia.

"You're strong. But I could snap my fingers, and you'd all cease to exist." - Thanos, Avengers: Infinity War

#	1	2	3 A	3 B	4	5	6	7	Entregue Hojas:
Corrección	B	B-	B	B	B-	B	B-	B	Total
Puntos	15/15	13/15	10/10	10/10	10/15	15/15	7/10	4/10	84 +30 /100

Nombre: CAMILA BOTTARO
Padrón: 101053
Corregido por: NATI

114

1) (***) Tenemos información sobre recetas en 3 RDD de Spark.

Recetas (id_receta, nombre, tiempo_preparacion, dificultad)

Ingredientes (id_ingrediente, nombre)

Ingredientes por Receta (id_receta, id_ingrediente, cantidad)

Se pide:

a) Obtener el nombre de todas las recetas que tengan Cordero. (7 puntos)

b) Calcular la cantidad total de cada ingrediente si queremos hacer todas las recetas con Cordero que sean dulces. (8 puntos)

2) (***) Dada la exitosa convocatoria de los Juegos Olímpicos de la Juventud por parte del público, sus organizadores realizan distintos análisis para planificar las jornadas finales del certamen. Es por ello que cuentan con información en los siguientes archivos csv:

eventos.csv (id_evento, fecha, id_locacion, nombre_evento, id_categoria_deportiva, cantidad_espectadores)

locacion.csv (id_locacion, nombre, capacidad, capacidad_extendida, sede, latitud, longitud)

categorias_deportivas.csv (id_categoria_deportiva, nombre, año_de_adopcion)

El primer archivo cuenta con información de los eventos, indicando la fecha (en formato "YYYY-mm-dd hh:mm:ss") el lugar donde ocurrió (id_locacion) y la cantidad de espectadores que asistieron. Además se aporta información sobre la categoría deportiva a la cual pertenece el evento.

Por otro lado se tiene información sobre las distintas locaciones en la sedes del certamen en las que ocurrieron los eventos. Contamos con información de su capacidad total de espectadores así como de su capacidad extendida (cuantos asientos extras se pueden brindar sobre la capacidad de la locación).

Se desea obtener:

a) Nombre de la sede que acumule la mayor cantidad de espectadores en eventos durante el certamen del 14 al 15 de octubre inclusive. Esto es de vital importancia para distribuir el merchandising oficial del evento, para las fechas 5 miles. (7 pts.)

b) Nombre del evento y nombre de la categoría deportiva de aquellos eventos cuya cantidad de espectadores superó la **capacidad de la locación, más allá de la capacidad extendida**. Esto es de vital importancia para detectar problemas de seguridad o si es necesario realizar algún cambio de locación. (8 pts)

3) Resolver 2 (dos) y solo 2 de los siguientes ejercicios a elección (si resuelve más de 2 el ejercicio vale 0 puntos, sin excepciones). En cada caso indicar V o F justificando adecuadamente sus respuestas. Si no justifica vale 0 puntos sin excepciones.

a) Sea un archivo que contiene cinco millones de dígitos de Pi, no se puede saber si es random porque K(X) es intractable. (*) (10 pts)

b) Para poder suponer que un archivo es random debe verificarse que la entropía de Shannon sea máxima (*) (10 pts)

c) Las tablas de frecuencias de los compresores diccionarios de orden 3 o superior pueden ocupar tanto espacio que la compresión termina siendo ineficiente. (*) (10 pts)

d) Solo con compresores aritméticos podemos alcanzar la longitud ideal indicada por la entropía ya que permiten codificar un mensaje en cantidades no enteras de bits. (*) (10 pts)

4) Desafortunadamente, tenemos un set de datos con muchos puntos y necesitamos utilizar LSH para buscar los puntos más cercanos. Contamos con el siguiente set: {22,14,30,12} y las siguientes 4 funciones de hashing: h1(x)=(3x mod 7) mod 4, h2(x) = (2x mod 7) mod 4, h3(x)=((2x+2) mod 7) mod 4 y h4(x) = ((x+3) mod 7) mod 4. Se pide:

a. Usando b=2 y r=2, indique cómo quedan las tablas

b. ¿Cuáles puntos deberíamos comparar si nuestro query es el 16? Explique

c. ¿Qué podríamos hacer para reducir la cantidad de falsos negativos? ¿Y si quisiéramos reducir la cantidad de falsos positivos? (***) (15pts)

5) (**) Se tienen los siguientes documentos:

D1: CORDERO SAL PIMIENTA ROMERO

D2: CERDO CORDERO SAL CORDERO

D3: SAL CERDO LIMON

D4: CORDERO ENTRAÑA

D5: PIMIENTA PAPA CORDERO PIMIENTA

D6: CORDERO CORDERO CORDERO CORDERO

Dada la consulta 'CORDERO PIMIENTA' dar el resultado de la consulta rankeada utilizando TF IDF. (10 pts)

Considerando como relevante los documentos que no tengan otra carne que no sea CORDERO, calcular la Precisión, Recall y F1 Score. (5 pts)

6) Se tiene una matriz muy grande donde cada fila representa una imagen de una cara. Se quiere aplicar algún algoritmo de reconocimiento facial utilizando la SVD.

a. ¿Cómo podemos determinar el valor de k (cant. de dimensiones a utilizar)? Justifique

b. ¿Se podría reducir el espacio que ocupe la matriz sin perder información?

c. Una vez obtenido k, ¿Cómo podemos reducir los puntos originales a k dimensiones?

d. Si ahora quisiera reconocer una imagen, ¿Cómo podría usar la SVD para ello? (***) (10pts)

7) El CO1 desea evaluar la aceptación de las nuevas categorías deportivas que se sumaron en el año 2018 a los Juegos Olímpicos de la Juventud. Para ello es necesario que nuestra área de análisis de datos prepare una visualización que muestre a lo largo del tiempo de duración del certamen como fue evolucionando la cantidad de público que han tenido estas nuevas categorías.

Para desarrollar el punto debe partir como base de la información que cuenta en el punto 2, ampliando con otras posibles fuentes de datos, el contenido de la misma. (***) (10 pts)

Camila Bojman 101055

2/6

HOJA N°

FECHA

① a)

```
rdd_ingrediente_receta = ingredientes_por_receta.map(lambda x: (x[1], x[0]))   # (id_ing, id_rec)

rdd_recetas_cordero = ingrediente_receta.join(ingredientes) -> (id_ing, (id_rec, nombre))
                        .filter(lambda x: x[1][1] == 'cordero')
                        .map(lambda x: (x[1][0], 1))

rdd_recetas = receta.map(lambda x: (x[0] x[1]))   # (id_receta, nombre)
                                                  # Lo(id_receta, 1)

rdd_nombres : rdd_recetas_cordero.join (recetas)   # (id_receta, (1, nombre)) <- rdd
                                                   # es inner = left.
                                                   # x q nota nada en recetas q' no este en rdd_re
                        .map(lambda x: (x[1][1]))
                        .collect()
```

~~rdd_n~~

✓

b) recetas_faciles = receta.filter(lambda x: x[3] == 'facil')
.map(lambda x: (x[0], 1)) -> (id_rec, 1)

(id_rec (id_ing, 1))

.map(lambda x: (x[1][:]))

~~rdd_n~~ .collect()

b) recetas_faciles = receta.filter(lambda x: x[3] == 'facil')
.map(lambda x: (x[0], 1)) → (id_rec, 1)

~~rdd_ingrediente_receta = ingrediente_por_receta.mapt~~ (id_rec, (id_ing, #))
(x[0], (x[1], x[2]))

rdd_ing_receta = ingredientes_por_receta.map(lambda x: ~~(x[0], (x[1], x[2]))~~)

rdd_faciles_ing = rectas_faciles.join(rdd_ing_receta) → (id_rec, (1, (id_ing, #)))

(id_ing, #) ← .map(lambda x: (x[1][1][0], x[1][1][1]))
.reducebykey(lambda x, y: x + y)

rdd_cant = rdd_faciles_ing.join(ingredientes) → id_ing, (#, nombre)
.map(lambda x: (x[1][1], x[0][0]))
.collect()

✓

(2)

a) df_eventos = pd.read_csv('eventos.csv')
df_locacion = pd.read_csv('locacion.csv') Ojo con esto. podrías convertirla a fecha o sino

df_eventos_fechas = df_eventos.loc[(df_eventos['fecha'] = '2018-10-14') | (df_eventos['fecha'] = '2018-10-15')]

usar un str.contains porque el formato tiene hora también

df_ev_loc = df_eventos_fechas.join(locacion, on = 'id_locacion')

sede_max_esp = df_ev_loc.groupby('sede')['cantidad_espectadores']
.sum()
.to_frame()
.rename(colums={'cantesp': 'total_espect'})
.reset_index()
.sort_values('total_esp', ascend=False)
.head(1)

sede_max_esp.

o nlargest para no hacer un sort y quedarnos con 1

b)

df_locacion['cap_maxima'] = df_locacion['capacidad'] *

cant_esp.

b)

para no hacer un sort y pvedarnos con 1

df_locacion['cap_maxima'] = df_locacion['capacidad'] + df_locacion['capacidad_extendida']

df_locacion.join('eventos', ON='id_locacion')

df = df.loc[df['capacidad_maxima'] < df['cant_esp']:]
df = df[['nombre_evento', id_categoria_deportiva']]

df_categoria = pd.readcsv('categoria_deportiva.csv')
df_supera_cap = df.join(df_categoria, ON="id_categoria_deportiva
df_supera_cap = df_supera cap[['nombre_evento', 'nombe']]

NOTA

df_supera_cap.

Camila Beiman 101055.

3/6

HOJA N°

FECHA

(3) a) Falso: [redacted], $K(x)$ es intractable y, por lo tanto, no se puede saber si un archivo es random. A pesar de eso existen algoritmos para generar el número pi que ocupan menos espacio que el número con un millon de dígitos en sí. Por lo tanto se puede afirmar que el archivo no es random ✓

b) Verdadero: No se puede verificar que un archivo sea random pero dado que la entropía de shannon es lo que mejor asemeja $K(x)$ la primer. pista para poder suponer que un archivo es random es que verifique que la entropía de Shannon sea alta. ✓

Camila Bejman 101055

6/6.

(4)

	h_1	h_2	h_3	h_4
22	3	2	3	∅
14	∅	∅	1	3
10	2	2	∅	2
12	1	3	∅	1

or (22,14) and ∅.

or (10,12).

✓

c) Para reducir la cantidad de falsos positivos se puede aumentar r. Para reducir la cantidad de falsos negativos se puede aumentar b. O, en este caso en particular, hacer primero los ands y luego los ors. ✓

a) Al hacer dos ors, vamos a generar que los dos puntos dentro del mismo bucket colisionen.
Y luego al aplicar dos ands, vamos a obtener un conjunto vacío dado que los buckets elejidos al momento de hacer los ors necesariamente son disjuntos. Es decir, no encontraríamos dos puntos cercanos.

(b) no resuelve

Camila Bejman - 101055

4/6

(5)

Q: 'cordero pimienta'

$IDF_{cord} = \log\left(\frac{6+1}{5}\right) = 0.146$

$IDF_{pimienta} = \log\left(\frac{6+1}{2}\right) = 0.544$

	D_1	D_2	D_3	D_4	D_5	D_6
cordero	0.146	0.292	0	0.146	0.146	0.584
pimienta	0.544	0	0	0	1.088	0
$TF\text{-}IDF_q$	0.69	0.292	0	0.146	1.234	0.584

$IDF_{palabra}$

$TF_{palabra, D_i}$

resultado:

D_5	1°
D_1	2°
D_6	3°
D_4	4°
D_2	5°

✓

Relevantes: $\{D_1, D_5, D_6\}$ $|Relevantes| = 3$

Recuperados: $\{D_1\ D_2\ D_4\ D_5\ D_6\}$ $|Recuperados| = 5$

Relevantes Recuperados: $\{D_1\ D_5\ D_6\}$. $|Relevantes\ Recuperados| = 3$

Precisión = $\frac{|Rel\ Rec|}{|Rec|} = \frac{3/5}{}$

Recall = $\frac{|Rel\ Rec|}{|Rel|} = 3/3 = \underline{1}$

$F_\beta score = \frac{(\beta^2+1)\,PR}{\beta^2 P + R}$ ∴ $F_1\ score = \frac{2 \cdot 3/5 \cdot 1}{8/5} = \underline{3/4}$

✓

(6)

a) Hay dos formas de seleccionar h_m. Por un lado se puede calcular el porcentaje de energía ignorado de nuestro set de datos ✓

es.

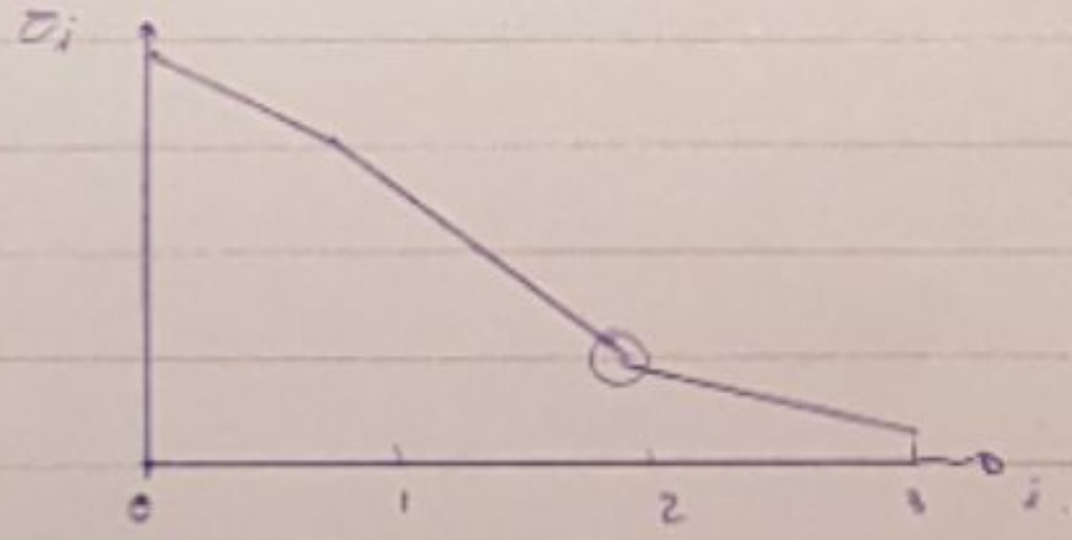
Esto se logra calculando la energia que obtenemos al reducir nuestro set de datos a h_m dimensiones sobre la energia original Λ_{total} de nuestro set de datos.

$$\frac{\sum_{i=0}^{h_m} \sigma_i^2}{\sum_{i=0} \sigma_i^2}$$

donde n = # valores singulares.
$\ni \sigma_i > \sigma_{i+1} \quad \forall i \in (0, n-1)$

De esta forma podemos probar con distintos valores de h_m hasta obtener una ~~por proporcion - reducciones~~ ~~cantidad~~ dimensión adecuada.

Por otro lado se puede generar un grafico de lineas donde en el eje X se encuentren los valores singulares y en el eje y el valor que toman. Por ejemplo si tenemos $\sigma_0 = 8 \quad \sigma_1 = 7 \quad \sigma_2 = 2 \quad \sigma_3 = 1$



✓

Luego de observar el gráfico buscamos los 'codos' (en nuestro ejemplo se genera en σ_2) y con ellos nos orientamos para ~~el~~ elejir un h_m

Camila Bejman 101055

5/6

HOJA N°

FECHA

⑥ b) El espacio

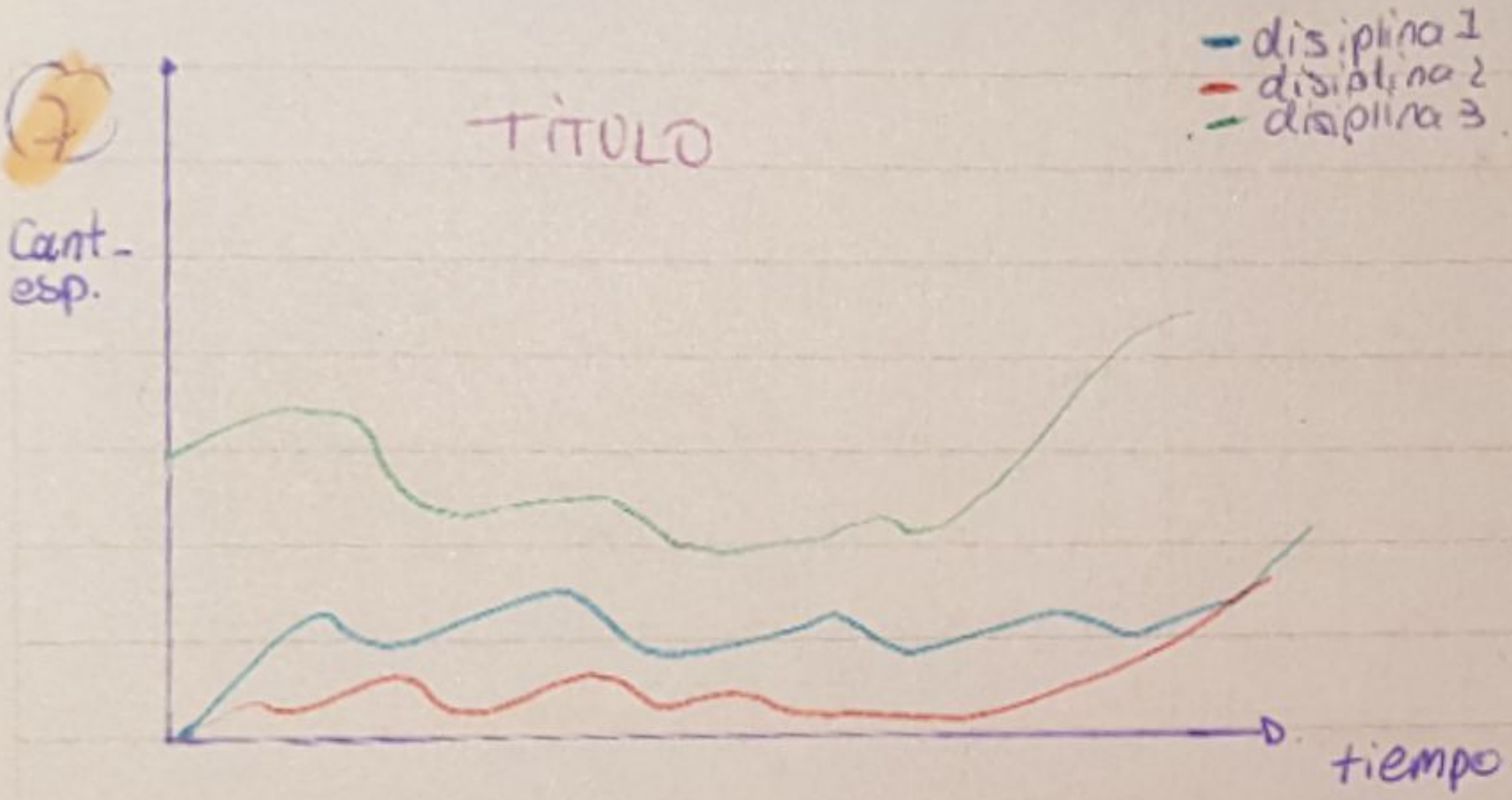
La máxima reducción que se puede generar con una DVS es r, siendo r el rango de la matriz. Por lo tanto el espacio que ocupa la matriz se puede reducir sin perder información siempre y cuando tenga al menos un valor singular nulo. ✓

sin perder información ✓

c) Una vez obtenido k la reducción es tan simple como conservar las k primeras columnas de U.

d) no resuelve

(7)



El grafico es una representación lineal de cómo se va modificando la cantidad de espectadores de las nuevas disiplinas.

No explico por qué elije este tipo de visualización y no otro, ...