Organización de Datos 75.06. Primer Cuatrimestre de 2020. Examen parcial, segunda oportunidad:

Importante: Lea bien todo el enunciado antes de empezar. Para aprobar se requiere un mínimo de 60 puntos (60 puntos = 4) con al menos 30 puntos entre los ejercicios 1 y 2. Si tiene dudas o consulta estaremos disponibles vía meet, pero tengan en cuenta que solo se contestarán dudas de enunciado, y no deben compartir por esa vía nada relacionado con la resolución. Está prohibido realizar cualquier actividad que pueda molestar a los demás. El criterio de corrección de este examen estará disponible en forma pública en el grupo de la materia.

"I believe in second chances. I don't believe in third chances" – Doran Martell, Game oThrones

#	1	2	3	4	Entrega Hojas:
Corrección					Total:
Puntos	/30	/25	/25	/20	/100

Nombre:
Padrón:
Corregido por:

1) Dado los acontecimientos en USA, deseamos obtener datos que nos den mayor información sobre las muertes de gente de raza negra por parte de oficiales de policía.

Para ello, tenemos un csv con información sobre las muertes por parte de oficiales de policía en USA para 2015 hasta 2017:

`(name, date, race, city, state)`

Y otro csv con información sobre el porcentaje de pobreza en las ciudades de USA:

`(state, city, poverty_rate)`

Se pide:

a) Obtener las 10 ciudades con mayor diferencia entre el porcentaje de pobreza de la ciudad y el porcentaje de pobreza del estado en el que se encuentra esa ciudad. Por ejemplo si en la ciudad de Houston la pobreza es de 15.2 y la pobreza en Texas (el estado donde se encuentra Houston) es de 11.1, la diferencia es 4.1.(15 pts)

b) Obtener la cantidad de muertes de gente de raza negra por parte de oficiales de policía, agrupada por estados que compartan el mismo nivel de pobreza redondeado al entero más cercano. Por ejemplo, si NJ tiene una pobreza de 10.33, AL una de 20.64 y AZ una de 10.44, NJ y AZ quedarían juntos representados por el nivel de pobreza de 10 y AL en otro grupo con el nivel 21. La salida debe tener el formato: `(nivel_de_pobreza, total_de_muertes)` (15 pts)

Resolver los puntos usando la API de RDDs de PySpark. (30 pts)

2) Un importante servicio de monitoreo de aplicaciones cloud, sufrió un incidente en las últimas semanas, tras el cual está conduciendo una investigación del impacto que continua teniendo en sus integraciones a nivel de conectividad. Como analista externos, tu misión es:

a) Calcular las métricas sumarizadas `'handshake_failed'`, `'ssl_failed'`, `'tls_failed'` para cada uno de los `integration_id` a partir del timestamp '1592187602', momento en el cual comenzó el incidente. El formato esperado del df salida es:

	handshake_failed	ssl_failed	tls_failed
integration_1	tot_handshake_1	tot_ssl_1	tot_tls_1
integration_2	tot_handshake_2	tot_ssl_2	tot_tls_2
...	...	...	...
integration_n	tot_handshake_n	tot_ssl_n	tot_tls_n

b) Indicar aquellas 5 integraciones que tuvieron la mayor cantidad de errores `'handshake_failed'` a partir del inicio del incidente. El formato esperado del df de salida es: `(integration_name, tot_handshake_failed)`

Considerando los siguientes dataframes en CSV:

- metrics.csv describiendo información de métricas obtenidas. El mismo tiene el formato `('client_id', 'integration_id' ,'metric', 'timestamp', 'value')` donde la columna `'metric'` indica el nombre de una métrica que se registra en un cierto momento (`'timestamp'`) cuyo valor se guarda en la columna `'value'`.

- integrations.csv describiendo las integraciones posibles que da a sus clientes el servicio de monitoreo. `('integration_id', 'data_provider', 'integration_name')`

Resolver ambos puntos utilizando Pandas (25pts)

3) Dados los siguientes vectores en 7 dimensiones:

V1 = [0, 1, 0, 1, 1, 0, 0]
V2 = [1, 0, 1, 0, 1, 0, 1]
V3 = [1, 1, 1, 0, 0, 1, 0]
V4 = [0, 1, 0, 0, 1, 0, 1]
V5 = [1, 1, 1, 1, 1, 1, 0]
V6 = [1, 1, 1, 1, 0, 0, 0]
V7 = [0, 0, 0, 0, 0, 0, 1]
V8 = [0, 0, 0, 1, 1, 1, 0]

Se utiliza un minhash para la distancia de hamming usando los bits 0, 1 y 5. Es decir que tendremos r = 3 y b = 1. Decidimos usar para LSH una tabla de hashing de 6 registros con buckets de tamaño 4.

a. Realice el pre-procesamiento necesario para poder usar LSH con los vectores. Indique precisamente qué estructura tendrá la tabla, qué se almacena en la misma y de qué forma.
b. Queremos hallar el vector más cercano a Q = [1, 0, 1, 0, 0, 0, 1]. Explique de qué forma encontraría dicho vector a partir de la estructura construida en el punto **a**.
c. La tabla usada para este ejercicio tiene capacidad para 24 registros que es el triple de los registros que almacenamos (8). Indique de qué forma podría reducirse la cantidad de espacio usada.

(25pts)

4) Una empresa de publicidad cuenta con datos que reflejan la cantidad de clicks que los usuarios han hecho sobre avisos en diferentes ciudades. La empresa trabaja en 17 ciudades del mundo y realizó la prueba con 1500 avisos durante 30 días en cada ciudad obteniendo como resultado una matriz de 17x1500 en donde cada celda indica la cantidad de veces que el aviso fue clickeado en cada ciudad.

a) La empresa quiere agrupar a los avisos en 4 categorías usando para ello la SVD como método de reducción de dimensiones de la matriz original.
Indique de qué forma procedería a partir de los datos originales. Explique en cada paso qué haría y qué resultados obtendría. (10pts)

b) La empresa cuenta con un presupuesto de U$S 100.000 para invertir y necesita saber cuánto invertir en cada ciudad y cómo. Indique de qué forma repartiría el presupuesto entre las ciudades y las categorías obtenidas en el punto a.
¿Deberían los directivos de la empresa preocuparse acerca de la escalabilidad de la solución planteada? (10pts)