1. Se tiene un problema de KDD donde se espera conocer cuál será el valor en pesos de una acción que cotiza en la bolsa de Buenos Aires. La base de datos que se dispone describe con 200 atributos la variación de dicho precio. Responda:
    a. ¿Qué tarea de data mining se utilizará para modelar el precio? ¿Por qué? ¿Qué tipo de aprendizaje es?
    b. ¿Qué estrategia de holdout se debería aplicar? ¿De qué depende que funcione la estrategia propuesta?

Utilizaria una estrategia mediante arboles de decision, ya que tengo la variable objetivo(precio) que quiero predecir y un conjunto de entrenamiento, este tipo de aprendizaje es supervisado. Antes de utilizar el algoritmo deberia categorizar la variable objetivo (posiblemente mediante binning) y reemplazar faltantes en el dataset (analizar outliers, etc).
Utilizaria 2Folds dividiendo el datasets en 75 training y 25 testing.
Dependeria de tener una buena variabilidad en el conjunto de entrenamiento para poder aprender de los ejemplos y categorizar correctamente la variable objetivo.

2. Explique someramente cómo funciona el clustering jerárquico. ¿Cómo funciona un *linkage* completo?

En clustering jerarquico cada observacion es un cluster, en cada iteracion se juntan los 2 clusters mas cercanos hasta llegar a uno solo.
En el caso de linkage completo se compara calcula la distancia usando todas las observaciones de cada cluster(como un producto cartesiano), esto se traduce en un dendograma mas facil de analizar y encontrar el mejor punto de corte para elegir los clusters.

3. Dado el siguiente conjunto de transacciones y con un *min_sup = 0.75* y *min_conf = 0.75*, responder:

TID	Items
1	KYF
2	ZYQ
3	KZYQ
4	ZQ
5	KZYQ

A. ¿Cuales son los 1-itemset frecuentes?
B. ¿{Z, Q} y {Z, Y, Q} son frecuentes? En el caso de no ser itemsets frecuentes alguno de ellos explique cuáles son las razones.
C. ¿{ZQ} =>{Y} y {Z} =>{Q} son reglas interesantes? Justifique.

Itemset /regla	soporte	confianza
K	3/5 = 0.6	
Y	4/5 = 0.8	
F	1/5 = 0.2	
Z	4/5 = 0.8	
Q	4/5 = 0.8	
ZQ	4/5 = 0.8	
ZYQ	3/5 = 0.6	
ZQ → Y	3/5 = 0.6	3 / 4 = 0.75
Z → Q	4/5 = 0.8	4 / 4 = 1

a) Los 1-itemsets frecuentes son **Y, Z** y **Q.**

b) el itemset **ZQ** es frecuente ya q tiene el soporte minimo mientras que el itemset **ZYQ** no lo alcanza

c) **Z → Q** es interesante ya que pasa la confiaza y soporte minimo, mientras que **ZQ → Y** no llega al soporte minimo

4. Dados los siguientes centroides C1 = (12, 16) y C2 = (15,9):

Precio	Fallas
11	17
13	16
13	14
13	17
12	17
11	15
15	5
16	12
14	11

A. Indique a qué centroide pertenece cada instancia.
B. ¿Cuales son los nuevos centroides de la asignación realizada en A?
C. ¿Considera que el agrupamiento es adecuado? Justifique
D. ¿Qué puede concluir a partir de la interpretación de los clusters obtenidos?

Distancias de manhatan (mas facil la cuenta):

Precio	Fallas	Distancia a cluster C1	Distancia a cluster C2	Corresponde al cluster
11	17	|12 – 11| + |16 – 17| = 2	|15 – 11| + |9 – 17| = 12	C1
13	16	|12 – 13| + |16 – 16| = 1	|15 – 13| + |9 – 16| = 9	C1
13	14	|12 – 13| + |16 – 14| = 3	|15 – 13| + |9 – 14| = 7	C1
13	17	|12 – 13| + |16 – 17| = 2	|15 – 13| + |9 – 17| = 10	C1
12	17	|12 – 12| + |16 – 17| = 1	|15 – 12| + |9 – 17| = 11	C1
11	15	|12 – 11| + |16 – 15| = 2	|15 – 11| + |9 – 15| = 10	C1
15	5	|12 – 15| + |16 – 5| = 14	|15 – 15| + |9 – 5| = 4	C2
16	12	|12 – 16| + |16 – 12| = 8	|15 – 16| + |9 – 12| = 4	C2
14	11	|12 – 14| + |16 – 11| = 7	|15 – 14| + |9 – 11| = 3	C2

B) Los nuevos centroides son C1 = (73/6 = 12.16, 96/6 = 16) y C2 = (45/3 = 15, 28/3 = 9.3). Pasandolo en limpio los centroides son: C1(12, 16) y C2 (15, 9)

C) Considero que el agrupamiento es adecuado, para corroborarlo haria una grafica de silueta (si llego lo hago)

D) Entiendo que obtuve 2 clusters, en C1 hay elementos de precio ligeramente mas barato pero con mas fallas, mientras q en C2 hay elementos ligeramente mas caros pero con menos fallas.

5. Dado el siguiente conjunto de datos:

riesgo	seguridad	ingresos	deuda	historial de crédito
alto	baja	$0 a $15K	alta	malo
alto	baja	$15K a $35K	alta	desconocido
moderate	baja	$15K a $35K	baja	desconocido
alto	baja	$0 a $15K	baja	desconocido
bajo	baja	> $35K	baja	desconocido
bajo	adecuada	> $35K	baja	desconocido
alto	baja	$0 a $15K	baja	malo
moderate	adecuada	> $35K	baja	malo
bajo	baja	> $35K	baja	bueno
bajo	adecuada	> $35K	alta	bueno
alto	baja	$0 a $15K	alta	bueno
moderate	baja	$15K a $35K	alta	bueno
bajo	baja	> $35K	alta	bueno
alto	baja	$15K a $35K	alta	malo

a) Si utilizamos ID3 cuál sería el atributo que iría a la raíz del árbol si utilizo GI.
b) ¿Cambia la raíz si utilizo Gain Ratio?

a) iria el atributo con mayor Ganancia de Informacion

6. ¿Explique cuál es el problema de tener un árbol de decisión muy profundo? ¿Cómo se llama ese problema y cómo se puede resolver en ese contexto?

El problema de un arbol muy profundo es que pierde la habilidad de generalizar, este problema se llama overfitting ya que el arbol se sesga mucho al dataset de entrenamiento que recibio, la forma de solucionar el problema es podar el arbol, indicando de que no puede pasar de una X profundidad. Tambien se puede indicar que sus hojas tengan una cantidad minima de observaciones ya q sino tambien puede llegar a hacer overfitting .

7. Explique en qué consiste un RDD. ¿Qué ventajas aporta esta estructura de datos?

RDD es una “dataset” en la arquitectura de SPARK, el mismo contiene todos los datos y permite realizar operaciones sobre el mismo, devolviendo un valor u otro RDD, es la forma de utilizar SPARK ya que cuando se le indica a un RDD que realize cierta accion, SPARK por atrás se encarga de paralelizar las tareas en los distintos workers para llevarla a cabo

8. ¿Cómo calcularía el soporte de los ítemset del ejercicio 3 utilizando el framework MapReduce? Indique cuales serían cada uno de los pasos.

El paso de Map seria (suponiendo que recibo una transaccion entera) separar por 1-itemset y enviar cada uno en una clave-valor donde el itemset es la clave, y tiene un 1 por valor, despues armaria las diferentes combinaciones de 1-itemsets ordenadas alfabeticamente y las mandaria como otro par con el valor 1.

ej: recibo { D, A, F } y envio (D, 1), (A, 1), (F,1), (A-D, 1), (A-F, 1), (D-F, 1), (A-D-F, 1)

Por otro lado cuando terminan los mappers y se ordenan todos los pares, el reducer los recibe y va acumulando por cada clave el total de apariciones, lo unico que no se es como sabria el total de transacciones para despues hacer la division.

Se me ocurre como fix que el mapper a su vez envie un KV (“AAAAA-cant-transacciones”, 1) la cual tamb es sumada en el primer reducer y un segundo reducer calcula el soporte final para cada clave tomando el valor de la clave dividio el valor de la clave cant-transacciones.