PROMOCIONAL

ORGANIZACIÓN DE DATOS
75.06

Curso: Argerich

Franco Nicolás Batastini 103775

1) Sean los siguientes puntos en dos dimensiones: x1=(0,0), x2=(8,0), x3=(16,0), x4=(0,6), x5=(8,6), x6=(16,6). Sobre estos puntos queremos usar K-Means, con la distancia euclidea, para encontrar 3 clusters. Sabemos que el resultado del algoritmo depende de la forma en la que elijamos los tres centroides iniciales. En base a esto responder:

a) ¿Cuántas configuraciones iniciales posibles existen? (5 pts)
b) ¿Cuántas clusterizaciones finales son posibles? (es decir aquellas clusterizaciones en las cuáles K-Means ya ha convergido). (5 pts)
c) ¿Cuál es el máximo número de pasos que podemos necesitar desde cualquiera de las inicializaciones iniciales posibles hasta cualquiera de las posibles clusterizaciones.? (10 pts)

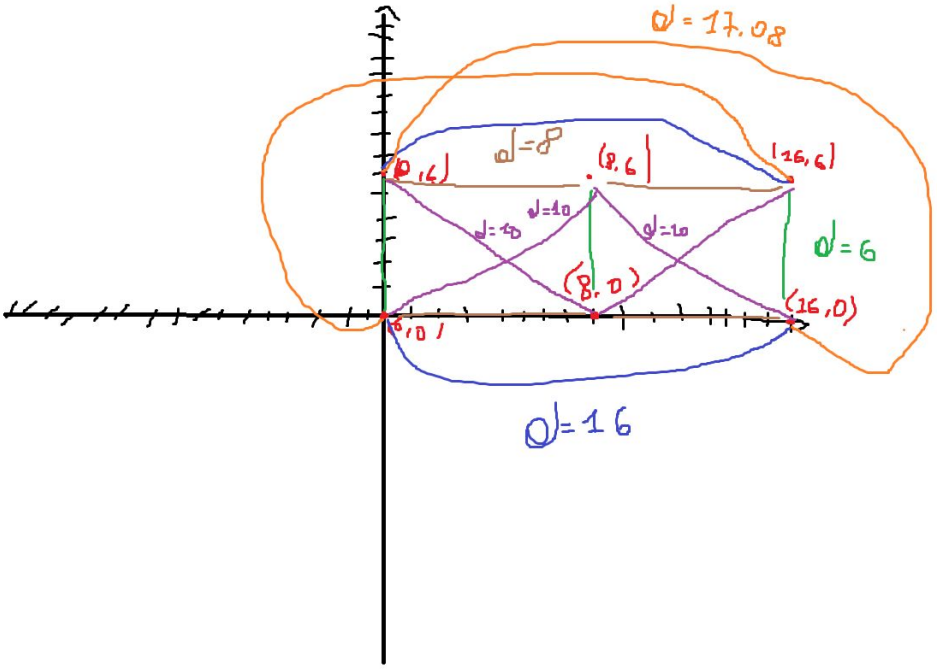
1) Se hará k-means con k=3 clusters

a) Las opciones posibles serán tomando todas las combinaciones posibles de 3 puntos tomados. Es decir:

- x1,x2,x3
- x1,x2,x4
- x1,x2,x5
- x1,x2,x6
- x1,x3,x4
- x1,x3,x5
- x1,x3,x6
- x1,x4,x5
- x1,x4,x6
- x1,x5,x6
- x2,x3,x4
- x2,x3,x5
- x2,x3,x6
- x2,x4,x5
- x2,x4,x6
- x2,x5,x6
- x3,x4,x5
- x3,x4,x6
- x3,x5,x6
- x4,x5,x6

Y luego de tomados los puntos iniciales, se toman las posiciones de ellos mismos como centroides, y cada punto restante que no haya sido elegido como centroide, será asignado al centroide más cercano (Distancia euclídea mínima) para generar el cluster inicial.

Viendo las distancias:



Se ve que si se toman los 3 puntos con coordenadas distintas en x, los clusters iniciales serán los que están de forma vertical, es decir:

C1 = [(0,6) ; (0,0)]
C2 = [(8,6) ; (8,0)]
C3 = [(16,6) ; (16,0)]

Si se toman 2 puntos que estén alineados verticalmente (Mismo X, distinto Y), y un 3ro que no esté entre medio de estos y los demás puntos, se obtiene que los dos puntos verticales se juntan horizontalmente con los que estén en la línea vertical donde no está el 3er punto, por ejemplo:
Si los puntos iniciales son (0,6) ; (0,0) ; y (16,6), quedará:

C1 = [(0,6) ; (8,6)]
C2 = [(0,0) ; (8,0)]
C3 = [(16,6) ; (16,0)]

Y por último, está también el caso donde se toman dos puntos con coordenada en “Y” distinta, pero con mismo “X”, y estos puntos tienen coordenada x = 16, o coordenada x = 0, pero con el tercer punto con coordenada x=8, entonces los clusters quedan con los dos puntos de los extremos solos, y con el que está en coordenada x=8 tomando a todos los demás puntos. Ejemplificando con los puntos (16,6) ; (16,0) ; (8,0):

C1 = [(16,6)]
C2 = [(16,0)]
C3 = [(8,0) ; (8,6) ; (0,0) ; (0,6)]

Entonces, considerando que la cantidad de clusters iniciales posibles serán 1 para el primer caso (Todos Verticales) (8 combinaciones de puntos posibles), 4 para el segundo caso (8 combinaciones de puntos posibles), y 2 para el tercer caso (4 combinaciones de puntos posibles), entonces las combinaciones iniciales de clusters posibles totales serán 7.

Y combinaciones de puntos posibles iniciales serán 6C3 = 20, que se ven más detalladas al inicio de esta respuesta

b) Las opciones son las siguientes:

- Para el caso 1 del punto a, sucede que el cluster inicial es el cluster final, ya que los puntos están todos a distancia 3 de su centroide, y cualquier distancia a otro centroide (sqrt(8^2+3^2)=8,54) será mayor que la del cluster al cual ya están asignados , por lo que esta configuración inicial tiene esta única configuración final.
- Para el caso 2 del punto a, el cluster inicial tendrá distancia mínima hacia el centroide para la unión vertical, por lo que queda analizar las dos distancias horizontales. La distancia al centroide será de 4 para ambos casos, y los centroides estarán siempre a una distancia ||(8-4,6-0)|| = sqrt(4^2+6^2) = 7,21; la cual será mayor que la distancia horizontal que hay hacia los centroides actuales. De esta forma, la configuración inicial también será una posible configuración final del algoritmo de K-means.
- Para el caso 3 del punto a, el centroide de los puntos solos estará a una distancia 0 de los mismos, por lo que estos no cambiarán de cluster, y los demás puntos estarán con un centroide a distancia 5 en el cluster conjunto de 4 elementos. Como se sabe que los más propensos a cambiar serían los que están en el medio por tener menor distancia que los extremos, se compara la distancia de estos al punto, y la distancia de estos al centroide del cluster de 4 elementos.
  Como la distancia al centroide es 5, y la distancia a los puntos de los otros clusters (Que serán su centroide), es de 8, o de 10 (Detallado en el gráfico), el cluster inicial también será una configuración final para los mismos.

Por lo tanto, se puede decir finalmente, que la cantidad de configuraciones iniciales van a ser las mismas que la cantidad posible de configuraciones finales. Siendo esta de 7 al igual que en el punto a.

c) Como se dio más en detalle en el punto b a la hora de decidir la cantidad posible de clusters finales, se ve que las iteraciones necesarias para pasar de cualquier configuración inicial en este ejemplo hasta el cluster final, será de 0 iteraciones, ya que el cluster inicial siempre finaliza siendo el cluster final, y no hay forma de minimizar más la ecuación que busca minimizar k-means que es la siguiente:

$$\sum_{i=clusters} \sum_{j=\ puntos\ del\ cluster\ i} ||x_j - \mu_i||^2$$

Donde x_j es el punto en cuestión, y μ_i el centroide al cual se asigna ese punto.

En resumen, lo que busca K-means es disminuir la distancia de los puntos al centroide del cluster al cual pertenecen, y para este ejercicio, la cantidad de iteraciones necesarias para minimizar la ecuación, será de 0 iteraciones debido a que inicialmente ya la minimizan.

2) Dada la siguiente matriz que representa si los usuarios 1 a 6 les gustaron o no las series A-F:

		Items					
		A	B	C	D	E	F
Usuarios	1	Si	No		Si	Si	
	2	No			No	Si	Si
	3		No	No	Si		Si
	4	No		Si	No	No	
	5	Si	No	Si			Si
	6	No	Si	No		No	Si

Usar la semejanza de Jaccard y collaborative filtering user-user con k (cantidad de vecinos) = 3 para estimar la probabilidad de que al usuario 6 le guste la serie " D".

(20pts)

2)

DATOS

Cant vecinos = 3.
Semejanza = Jaccard

RESOLUCIÓN

Como la semejanza que se utiliza es la de Jaccard, la similaridad entre elementos se calcula como:

Intersección de elementos/Union de elementos = Semejanza de Jaccard

Y cómo busco los 3 vecinos más cercanos al usuario 6, realizare la semejanza entre todos los usuarios para con el usuario 6.

Sim(1 ; 6) = 0/6 = 0
Sim(2 ; 6) = 2/6
Sim(3 ; 6) = 2/6
Sim(4 ; 6) = 2/6
Sim(5 ; 6) = 1/5

Entonces, los 3 vecinos más cercanos al usuario 6 serán los usuarios 2, 3, y 4.

Ahora, basándome en la semejanza de los mismos, estimo el valor que le dará el usuario 6 a la serie D, tomando los "Si" como 1, y los "No" como 0. Si el resultado de realizar:

$$\frac{\sum Sim(6;i)\ .\ r_{i;D}}{\sum Sim(6;i)}$$

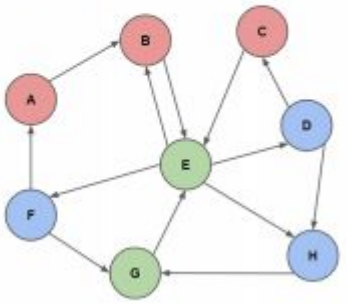
Da mayor o igual a 0,5 entonces se predice que al usuario 6 le gustará la serie, y si es menor a 0,5 entonces no le gustará:

$$\frac{2/6.\ 0 + 2/6\ .0 + 2/6\ .1}{2/6\ .\ 3} = \frac{1/3}{1} = \frac{1}{3}$$

Como 1/3 es menor a 0,5 entonces se predice que al usuario 6 no le gustará la serie D.

La probabilidad de que al usuario D le guste la serie sería de 1/3 , es decir, un 33,33%.

3) Dadas las siguientes páginas, sabemos que las mismas fueron asociadas con tres tópicos distintos (A, B, C en rojo para educación, D, F, H en azul relacionadas con economía y E, G en verde asociadas con tecnología), utilizar topic rank para calcular el ranking de cada una, sabiendo que se quiere favorecer las páginas relacionadas con economía, indicadas en el grafo con el color azul. (20 pts)



3)

Beta = 0,85 ← Dicho por Damian Martinelli en el Meet.

Al estar aplicando topic rank, se sabe que el vector de teletransportación que será aplicado al algoritmo de teletransportación para el page rank estará dado como 0 para los elementos que no son pertenecientes al tópico, y como 1/k, para los elementos pertenecientes al tópico, donde k es la cantidad de elementos que existen en ese tópico para el grafo.

Entonces, cómo se quiere favorecer a los que tienen “economía” como tópico, el vector de teletransportación quedaría como:

A	B	C	D	E	F	G	H
0	0	0	1/3	0	1/3	0	1/3

Debido a que D, F, y H son los elementos pertenecientes al tópico de economía, y cómo suman 3 elementos en total, será probabilidad 1/3 para cada uno.

Entonces ahora, puedo armar una matriz tal que me permite iterar de la siguiente forma:

$$Vector\ de\ Page\ Rank\ .\ Matriz\ =\ Vector\ de\ Page\ Rank\ nuevo$$

Donde se puede interpretar a la matriz como “Probabilidad de que la fila i en un random walk pase a la columna j”, y para cada columna direccionada por la fila, se calculará su probabilidad como:

Si el nodo fila tiene una arista direccionada hacia el nodo columna:

$$\beta.(1/Cant.Aristas\ Salientes\ del\ nodo\ fila) + (1-\beta).\ Vector\ de\ Teletransportación\ para\ el\ item\ columna$$

Si no tiene una arista que lo direccione:

$$(1-\beta).Vector\ de\ Teletransportación\ para\ el\ ítem\ columna$$

Armando la matriz me queda (Todas las filas suman 1 dado que es estocástica):

	A	B	C	D	E	F	G	H
A	0	0,85	0	0,05	0	0,05	0	0,05
B	0	0	0	0,05	0,85	0,05	0	0,05
C	0	0	0	0,05	0,85	0,05	0	0,05
D	0	0	0,425	0,05	0	0,05	0	0,475
E	0	0,2125	0	0,2625	0	0,2625	0	0,2625
F	0,425	0	0	0,05	0	0,05	0,425	0,05
G	0	0	0	0,05	0,85	0,05	0	0,05
H	0	0	0	0,05	0	0,05	0,85	0,05

Entonces ahora iterando con la cuenta nombrada anteriormente, inicializando el PageRank de cada nodo inicialmente como 1/n, donde n es la cantidad de elementos en el grafo:

PAGE RANK INICIALES							
A	**B**	**C**	**D**	**E**	**F**	**G**	**H**
1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8

Iterando con Vector_PR_n . Matriz = Vector_PR_n+1

Se obtienen los siguientes resultados:

Iteración 1:

$$\left(\frac{1}{8}\ \frac{1}{8}\ \frac{1}{8}\ \frac{1}{8}\ \frac{1}{8}\ \frac{1}{8}\ \frac{1}{8}\ \frac{1}{8}\right)\cdot\begin{pmatrix} 0 & 0,85 & 0 & 0,05 & 0 & 0,05 & 0 & 0,05 \\ 0 & 0 & 0 & 0,05 & 0,85 & 0,05 & 0 & 0,05 \\ 0 & 0 & 0 & 0,05 & 0,85 & 0,05 & 0 & 0,05 \\ 0 & 0 & 0,425 & 0,05 & 0 & 0,05 & 0 & 0,475 \\ 0 & 0,2125 & 0 & 0,2625 & 0 & 0,2625 & 0 & 0,2625 \\ 0,425 & 0 & 0 & 0,05 & 0 & 0,05 & 0,425 & 0,05 \\ 0 & 0 & 0 & 0,05 & 0,85 & 0,05 & 0 & 0,05 \\ 0 & 0 & 0 & 0,05 & 0 & 0,05 & 0,85 & 0,05 \end{pmatrix} = (\ 0,0531\ \ 0,133\ \ 0,0531\ \ 0,0766\ \ 0,319\ \ 0,0766\ \ 0,159\ \ 0,130\ )$$

Iteración 2:

$$\left(\frac{17}{320}\ \frac{17}{128}\ \frac{17}{320}\ \frac{49}{640}\ \frac{51}{160}\ \frac{49}{640}\ \frac{51}{320}\ \frac{83}{640}\right)\cdot\begin{pmatrix}0 & 0{,}85 & 0 & 0{,}05 & 0 & 0{,}05 & 0 & 0{,}05\\ 0 & 0 & 0 & 0{,}05 & 0{,}85 & 0{,}05 & 0 & 0{,}05\\ 0 & 0 & 0 & 0{,}05 & 0{,}85 & 0{,}05 & 0 & 0{,}05\\ 0 & 0 & 0{,}425 & 0{,}05 & 0 & 0{,}05 & 0 & 0{,}475\\ 0 & 0{,}2125 & 0 & 0{,}2625 & 0 & 0{,}2625 & 0 & 0{,}2625\\ 0{,}425 & 0 & 0 & 0{,}05 & 0 & 0{,}05 & 0{,}425 & 0{,}05\\ 0 & 0 & 0 & 0{,}05 & 0{,}85 & 0{,}05 & 0 & 0{,}05\\ 0 & 0 & 0 & 0{,}05 & 0 & 0{,}05 & 0{,}85 & 0{,}05\end{pmatrix}=\left(0{,}0325\ \ 0{,}113\ \ 0{,}0325\ \ 0{,}118\ \ 0{,}294\ \ 0{,}118\ \ 0{,}143\ \ 0{,}150\right)$$

Iteración 3:

$$\left(\frac{833}{25600}\ \frac{289}{2560}\ \frac{833}{25600}\ \frac{1507}{12800}\ \frac{3757}{12800}\ \frac{1507}{12800}\ \frac{731}{5120}\ \frac{3847}{25600}\right)\cdot\begin{pmatrix}0 & 0{,}85 & 0 & 0{,}05 & 0 & 0{,}05 & 0 & 0{,}05\\ 0 & 0 & 0 & 0{,}05 & 0{,}85 & 0{,}05 & 0 & 0{,}05\\ 0 & 0 & 0 & 0{,}05 & 0{,}85 & 0{,}05 & 0 & 0{,}05\\ 0 & 0 & 0{,}425 & 0{,}05 & 0 & 0{,}05 & 0 & 0{,}475\\ 0 & 0{,}2125 & 0 & 0{,}2625 & 0 & 0{,}2625 & 0 & 0{,}2625\\ 0{,}425 & 0 & 0 & 0{,}05 & 0 & 0{,}05 & 0{,}425 & 0{,}05\\ 0 & 0 & 0 & 0{,}05 & 0{,}85 & 0{,}05 & 0 & 0{,}05\\ 0 & 0 & 0 & 0{,}05 & 0 & 0{,}05 & 0{,}85 & 0{,}05\end{pmatrix}=$$

$$\left(0{,}0500\ \ 0{,}0900\ \ 0{,}0500\ \ 0{,}112\ \ 0{,}245\ \ 0{,}112\ \ 0{,}178\ \ 0{,}162\right)$$

Iteración 4:

$$\left(\frac{25619}{512000}\ \frac{92191}{1024000}\ \frac{25619}{512000}\ \frac{115069}{1024000}\ \frac{62713}{256000}\ \frac{115069}{1024000}\ \frac{45509}{256000}\ \frac{166307}{1024000}\right)\cdot\begin{pmatrix}0 & 0{,}85 & 0 & 0{,}05 & 0 & 0{,}05 & 0 & 0{,}05\\ 0 & 0 & 0 & 0{,}05 & 0{,}85 & 0{,}05 & 0 & 0{,}05\\ 0 & 0 & 0 & 0{,}05 & 0{,}85 & 0{,}05 & 0 & 0{,}05\\ 0 & 0 & 0{,}425 & 0{,}05 & 0 & 0{,}05 & 0 & 0{,}475\\ 0 & 0{,}2125 & 0 & 0{,}2625 & 0 & 0{,}2625 & 0 & 0{,}2625\\ 0{,}425 & 0 & 0 & 0{,}05 & 0 & 0{,}05 & 0{,}425 & 0{,}05\\ 0 & 0 & 0 & 0{,}05 & 0{,}85 & 0{,}05 & 0 & 0{,}05\\ 0 & 0 & 0 & 0{,}05 & 0 & 0{,}05 & 0{,}85 & 0{,}05\end{pmatrix}=$$

$$\left(0{,}0478\ \ 0{,}0946\ \ 0{,}0478\ \ 0{,}102\ \ 0{,}270\ \ 0{,}102\ \ 0{,}186\ \ 0{,}150\right)$$

Iteración 5:

$$\left(\frac{1956173}{40960000}\ \frac{1937167}{20480000}\ \frac{1956173}{40960000}\ \frac{2090121}{20480000}\ \frac{1106581}{4096000}\ \frac{2090121}{20480000}\ \frac{7610611}{40960000}\ \frac{1227283}{8192000}\right)\cdot\begin{pmatrix}0 & 0{,}85 & 0 & 0{,}05 & 0 & 0{,}05 & 0 & 0{,}05\\ 0 & 0 & 0 & 0{,}05 & 0{,}85 & 0{,}05 & 0 & 0{,}05\\ 0 & 0 & 0 & 0{,}05 & 0{,}85 & 0{,}05 & 0 & 0{,}05\\ 0 & 0 & 0{,}425 & 0{,}05 & 0 & 0{,}05 & 0 & 0{,}475\\ 0 & 0{,}2125 & 0 & 0{,}2625 & 0 & 0{,}2625 & 0 & 0{,}2625\\ 0{,}425 & 0 & 0 & 0{,}05 & 0 & 0{,}05 & 0{,}425 & 0{,}05\\ 0 & 0 & 0 & 0{,}05 & 0{,}85 & 0{,}05 & 0 & 0{,}05\\ 0 & 0 & 0 & 0{,}05 & 0 & 0{,}05 & 0{,}85 & 0{,}05\end{pmatrix}=$$

$$\left(0{,}0434\ \ 0{,}0980\ \ 0{,}0434\ \ 0{,}107\ \ 0{,}279\ \ 0{,}107\ \ 0{,}171\ \ 0{,}151\right)$$

Iteración 6:

$$\left(\frac{35532057}{819200000}\ \frac{160569267}{1638400000}\ \frac{35532057}{819200000}\ \frac{35195877}{327680000}\ \frac{114249503}{409600000}\ \frac{35195877}{327680000}\ \frac{17481389}{102400000}\ \frac{247043499}{1638400000}\right)\cdot\begin{pmatrix}0 & 0{,}85 & 0 & 0{,}05 & 0 & 0{,}05 & 0 & 0{,}05\\ 0 & 0 & 0 & 0{,}05 & 0{,}85 & 0{,}05 & 0 & 0{,}05\\ 0 & 0 & 0 & 0{,}05 & 0{,}85 & 0{,}05 & 0 & 0{,}05\\ 0 & 0 & 0{,}425 & 0{,}05 & 0 & 0{,}05 & 0 & 0{,}475\\ 0 & 0{,}2125 & 0 & 0{,}2625 & 0 & 0{,}2625 & 0 & 0{,}2625\\ 0{,}425 & 0 & 0 & 0{,}05 & 0 & 0{,}05 & 0{,}425 & 0{,}05\\ 0 & 0 & 0 & 0{,}05 & 0{,}85 & 0{,}05 & 0 & 0{,}05\\ 0 & 0 & 0 & 0{,}05 & 0 & 0{,}05 & 0{,}85 & 0{,}05\end{pmatrix}=$$

$$\left(0{,}0456\ \ 0{,}0961\ \ 0{,}0456\ \ 0{,}109\ \ 0{,}265\ \ 0{,}109\ \ 0{,}174\ \ 0{,}155\right)$$

Los resultados entre la iteración 6, y la iteración 5 distan de algunos pocos decimales, por lo que puedo decir que el algoritmo ya convergió, y los page rank finales son:

PAGE RANK FINALES							
A	**B**	**C**	**D**	**E**	**F**	**G**	**H**
0,05	0,10	0,05	0,11	0,26	0,11	0,17	0,15

4) Sabemos que tenemos una construcción de Count-Min de 3 filtros (F1, F2 y F3) de los cuales conocemos dos

F1 = [4,0,8,0,0,0,5,3]

F2 = [0,0,3,3,4,4,6,0]

Dados los siguientes candidatos para F3, cuál es el correcto y por que. (5 pts)

A = [5,0,1,3,2,4,4,1]

B = [5,4,1,0,1,6,0,4]

Considerando el F3 válido, cuál sería el resultado de la función de hashing para lograr la mayor estimación posible con los filtros F1, F2 y F3 para un cierto elemento "X" y cuál sería su resultado. (15 pts)

4) Para que el filtro sea adecuado, la sumatoria de las componentes del mismo deben ser iguales:

Para F1 se obtiene:

4+8+5+3 = 20

Para F2 se obtiene:

3+3+4+4+6 = 20

Por lo tanto para F3 debe ser 20 el valor de la suma de sus componentes.

Analizo el primer filtro A:

5+1+3+2+4+4+1 = 20

Y para el segundo filtro B:

5+4+1+1+6+4 = 21

Por lo que como él filtro A da igual que F1, y F2; y B no da lo mismo; descarto al filtro B, y el filtro correcto será A.

Considerando a F3 = A como válido, para lograr la mayor estimación posible de un elemento "X", se debe obtener los valores más altos de los filtros, y quedarse con el mínimo de estos (Debido a que el algoritmo funciona así), ya que si se toman los valores más altos de cada filtro, entonces se obtendrá el máximo count min de los mismos.

Entonces, tomamos de F1 el elemento 8, de F2, el elemento 6, y de F3=A el elemento 5.

Como 5 es el mínimo, presente en F3=A, entonces la máxima estimación posible para un dado elemento “X” será 5, y se tomará cuando la función de hashing colisione con el filtro 1 en la posición donde está 8 o 5, con el filtro 2 en la posición donde está el 6, y con el filtro 3, es decir, el filtro A, en la posición donde hay un 5. En resumen:

h1(x) = 2 o 6 ← Asociada a F1
h2(x) = 6 ← Asociada a F2
h3(x) = 0 ← Asociada a F3

Para lograr encontrar el máximo estimado, que será 5.

5) Tenemos información sobre precios de venta de propiedades en el país:

(fecha, tipo_propiedad, m2_totales, m2_cubiertos, habitaciones, dormitorios, baños, cocheras, provincia, ciudad, poblacion_ciudad, cant_escuelas_cercanas, servicios, estado_propiedad, antiguedad, precio)

Se quiere generar un modelo de predicción del precio de las propiedades usando XGBoost.

a) Indique el proceso de feature engineering que realizaría, indicando el detalle de las features que utilizaria, las tranformaciones que realizaria a cada columna y cuales datos agregaria. Indique una fila completa del set de datos final.

b) Indique qué cambios debería realizar al feature engineering si el modelo a utilizar es una red neuronal.

5)a)

Inicialmente hay que tomar en consideración que features tomamos como importantes para el precio de un inmueble. En este caso considero que:

- Fecha: No es sumamente importante; y mucho menos siendo que ya poseo la antigüedad de la propiedad. Por lo tanto este feature lo descarto.
- Tipo de Propiedad: Es necesario saberlo, pues no es lo mismo el valor de un departamento que el valor de una casa quinta, un campo, etc.
- Metros Cuadrados: Fundamental para el precio, ya que existe en reglas generales un precio por metro cuadrado construido, y un precio por metro cuadrado de lote.
- Habitaciones: También indispensable para conocer la cantidad de ambientes construidos.
- Dormitorios: Indispensable para saber cuántos de los ambientes son para dormir.
- Baños: También es un factor influyente a la hora de decidir en una propiedad por el precio.
- Cocheras: Añade valor la cantidad de cocheras presentes, por lo que también debe ser considerado.
- Provincia: Realmente necesario, ya que por ejemplo comprar en CABA es mucho más caro que comprar un departamento en Chaco, o en Jujuy, ya que son provincias de menor poder adquisitivo.
- Ciudad: También influye, no será el mismo precio en una ciudad urbana altamente poblada, que en una zona rural, de baja densidad poblacional.
- Población de la ciudad: Se podría utilizar acorde con la ciudad, utilizando algo del estilo mean encoding, ya que son dos variables altamente interrelacionadas entre sí.
- Cantidad de Escuelas cercanas: No lo considero altamente influyente en el precio, sino en la decisión de una persona para sus necesidades. Por lo tanto, lo descarto.
- Servicios: Una casa con servicios es más cara que una sin los mismos. Puede representarse como un OneHot encoding, para cada servicio que posea, spliteando los servicios 1 por 1 (Separar los strings por coma, y asignar 1 a cada servicio que aparezca en su columna).
- Estado de propiedad: Indispensable, pues una propiedad en deplorable estado es mucho más barata que una propiedad a estrenar.
- Antigüedad: Necesario para saber la infraestructura de la casa, y que es posible que posea internamente entre sus paredes. Además una casa con

más antigüedad, suele estar más desgastada que una a estrenar, y suele disminuir su precio.

Entonces, se descarta el feature de la fecha, se interrelacionan los features de densidad poblacional y ciudad, y se descarta el feature de escuelas cercanas.

Ahora que tengo un set de datos un poco más limpio, debo aplicar los encodings a las columnas categóricas. Los cuáles serán los siguientes:

- Tipo de Propiedad: Puedo hacer un one-hot encoding ya que no son muchos tipos (Chacras, quintas, departamentos, PH, etc. ; pero no superan los 15 tipos). Este encoding trata de hacer una columna binaria para cada tipo de propiedad presente en el dataset, donde es 1 si es de ese tipo, y 0 si no es de ese tipo.
- Provincia: Puede ser encodeado en base al mean encoding con smoothing del Target(Precio) para evitar el filtrado del mismo, pero permite dar una idea del precio que ronda en la provincia con el cual se está trabajando.
- Ciudad: Como se nombró anteriormente, se puede encodear como el promedio de la cantidad de habitantes para esa ciudad. Esto se puede hacer haciendo un df.groupby(“Ciudad”).agg(“poblacion_ciudad”:mean), y mergeandolo al dataset para posteriormente dropear las columnas ciudad y poblacion_ciudad.
- Servicios: Dado que dije que sería una columna con One Hot, se tomaría cada string de cada elemento, se lo splitearía por coma si fuesen comma separated values, y se harían las columnas de todos los servicios posibles, donde se pondría 0 si no posee dicho servicio, y 1 si posee el mismo.
- Estado de propiedad: Puede ser un One-Hot encoding considerándolo como Excelente, Bueno, Regular, Malo, Deplorable. Al ser 5 features, no es muy espacioso en memoria y puede ser útil y fácil de encodear.

El resto de los features no dropeados quedarían como vienen debido a que ya son numéricos y son utilizables de esa manera.

Entonces por cada fila del dataset final se presentan las siguientes columnas:

- OneHot_Tipo_De_Propiedad (Cantidad de columnas como tipos haya).
- Metros Cuadrados.
- Habitaciones.
- Dormitorios.
- Baños.
- Cocheras.
- Provincia_Encoding_Target.
- Ciudad_Encoding_Poblacion.
- OneHot_Servicios (1 por cada servicio posible).
- OneHot_Estado_Propiedad (5 columnas, 1 por cada estado posible).
- Antigüedad

5)b) Si el modelo fuese una red neuronal, sería recomendable que los valores se encuentren normalizados, ya que las redes neuronales trabajan mejor de esa manera.

Además una red neuronal requiere que no existan features categóricos, por lo que los encodings anteriores serían válidos, y útiles para este modelo luego de ser normalizados.

También es necesario que los datos que fuesen NULL o NaN sean dropeados, ya que la red neuronal falla con datos que sean NULL o NaN.