

7506-2020-1 Examen por Promoción

Tomas Rodriguez Dala

TOTAL POINTS

64 / 115

QUESTION 1

1 Arboles de Decisión 14 / 15

✓ - 1 pts Se pasa de listo

- 1 El enunciado dice CART no ID3. Comentario innecesario.

orden de la recomendación

QUESTION 7

7 Puntos Extra 0 / 15

✓ - 15 pts Sin puntos adicionales

QUESTION 2

2 Feature Engineering 10 / 15

✓ - 5 pts Deja atributos categoricos (no numericos) en el row final, Xgboost no maneja columnas que no sean numericas.

- 2 XGBoost no maneja atributos categóricos

QUESTION 3

3 Streaming 10 / 20

✓ - 10 pts Encuentra un buen valor para a y b pero no generaliza. Es decir no dice que valores de a y b funcionan bien o mal en general.

QUESTION 4

4 Clustering 0 / 15

✓ - 15 pts Sin hacer

QUESTION 5

5 Page Rank 15 / 15

✓ - 0 pts Correct

QUESTION 6

6 Recomendaciones 15 / 20

✓ - 5 pts Calcula el item-item contra todos en vez de contra los que ya adquirió A

- 3 No deberia calcular la semejanza contra todos los productos, solo contra los que compró A

- 4 A no compró P3, no deberia ser relevante en el

1) Para elegir el split debes calcular la gánonia de información de todos los splits posibles y elegir la menor. Calcula todos los ganancias:

~~H(0.5, 0.5)~~

$$H\left(\frac{3}{6}, \frac{3}{6}\right) = -\frac{1}{2} \log_2\left(\frac{1}{2}\right) - \frac{1}{2} \log_2\left(\frac{1}{2}\right) = 1$$

↑
proporción
de vegetales ↑
proporción de
carnívoros.

Primero debes ordenar los datos para ver mejor los splits:

11, C		• Split > 11
23, V		$H(" > 11") = H\left(\frac{3}{5}, \frac{2}{5}\right) = 0,971$
65, V		
80, C		$H(" \leq 11") = H\left(\frac{9}{11}, \frac{2}{11}\right) = 0$
85, C		$\Rightarrow GI = 1 - \left(\frac{5}{6} \cdot 0,971 + \frac{1}{6} \cdot 0\right) = 0,191$
100, V		

• Split > 23

$$H(" > 23") = H\left(\frac{2}{4}, \frac{2}{4}\right) = 1$$

$$H(" \leq 23") = H\left(\frac{1}{2}, \frac{1}{2}\right) = 1 \Rightarrow GI = 1 - \left(\frac{4}{6} \cdot 1 + \frac{2}{6} \cdot 1\right) = 0$$

- Split > 65

$$H(" > 65") = H\left(\frac{1}{3}, \frac{2}{3}\right) = 0,918 \Rightarrow GI = 1 - \left(\frac{3}{6} \cdot 0,918 + \frac{3}{6} \cdot 0,918\right) = 0,082$$

$$H(" \leq 65") = H\left(\frac{2}{3}, \frac{1}{3}\right) = 0,918$$

- Split > 80

$$H(" > 80") = H\left(\frac{1}{2}, \frac{1}{2}\right) = 1 \Rightarrow GI = 1 - \left(\frac{2}{6} \cdot 1 + \frac{4}{6} \cdot 1\right) = 0$$

$$H(" \leq 80") = H\left(\frac{2}{4}, \frac{2}{4}\right) = 1$$

- Split > 85

$$H(" > 85") = H\left(\frac{1}{1}, 0\right) = 0 \Rightarrow GI = 1 - \left(\frac{1}{6} \cdot 0 + \frac{5}{6} \cdot 0,971\right) = 0,191$$

$$H(" \leq 85") = H\left(\frac{2}{5}, \frac{3}{5}\right) = 0,971$$

Para decidir si hacer un split o no, el algoritmo tiene que tener en cuenta que la máxima ganancia de información posible en este split es de 0,191. Se puede lograr haciendo un split para mayor o igual a 11 o para menor a 85.

Como recordar que si el algoritmo es ID3 entonces no podría hacer un split en esta columna porque solo acepta splits sobre columnas numéricas.

1 Arboles de Decisión 14 / 15

✓ - 1 pts Se pasa de listo

- 1 El enunciado dice CART no ID3. Comentario innecesario.

2)

Como XGBoost es un árbol de decisión
hay que tener en cuenta que ~~no~~ puede
procesar atributos categóricos y numéricos.

Este ~~no~~ significa que no hace falta encodear
los features categóricos. (2)

Otra cosa es tener en cuenta en que
~~no~~ puede mirar un feature e lo vez
así que si hay combinaciones interesantes
hay que ~~no~~ agregarlos como features.

Hay que ir mencionando los cambios que le
haces a cada feature:

- **Fecha:** si la fecha viene en un string
la convierto a ~~string~~ un valor numérico, algo
como la cantidad de ~~string~~ ~~days~~ ~~since~~
días que pasaron desde 1/1/1800 (fecha donde
no había ningún año). ~~string~~ Como el valor
es numérico ahora el modelo puede hacer
~~mejores~~ splits.

- **Modelo y versión:** Las imágenes
en uno mismo ~~no~~ categóricos ~~modelos~~
"modelos - versión" porque ~~no~~ ~~modelos~~
~~modelos~~ ~~no~~ son valores ~~no~~ muy relacionados

No es lo mismo una versión 2.0 de un Fiat o un 208.

- Año: Lo dejaba como un valor numérico pero agregaría otra variable que sea "edad" que sea la diferencia entre la fecha de publicación y el año en que se hizo el auto para que el modelo pueda ver que tan viejo es el auto.
- Equipamiento: Agregaría Tanto fechar como equipamiento diferente porque donde nota whisms. Tiene un 1 si el auto tiene el equipamiento. Una vez que saque eso se habrá fijado los elementos en "equipamiento" a esa columna y borraría el fechar.

~~Más~~

Todo el resto de variables categoricas y numéricas las dejaría como están. Un ejemplo de un row sería:

- Fecha: 72980
 - Marca: Peugeot
 - Modelos - versión: 208 - 3.0
 - Años: 200³
- (No sé nada de autos, los valores son inventados).

- Contador de días desde la ~~velocidad del auto~~ ^{velocidad del auto}: 712
- Contador de vueltas: 3
- Segmento/Tomón: 3,65
- Aire acondicionado: 1
- Control de crucero: 0
- Audio Stereo: 1
- Kilómetro: 7567
- Estado: Usado
- Transmisión: Manual
- Color: gris
- Tipo de combustible: Gasolina
- Motor: 1.8T
- Potencia: 700
- Niño dueño: No
- Provincia: Córdoba.
- Dueño directo: Si
- Precio: ~~M\$~~ 800.000

2 Feature Engineering 10 / 15

✓ - **5 pts** Deja atributos categoricos (no numericos) en el row final, Xgboost no maneja columnas que no sean numericas.

- 2 XGBoost no maneja atributos categóricos

3) El algoritmo de Floyd - Martin funciona haciendo todos los valores que van entrando, convirtiéndolos en número binario y guardando el mayor número de ceros consecutivos a la derecha que sea (esta variable la llamaremos r). Para estimar el momento de orden 0 hace 2^r .

En este ejemplo los valores sencillos son $\{3, 1, 4, 5, 9, 2, 6\}$ o sea que tengo que elegir la función de hashing que haga que el algoritmo estime lo más cercano a 7 posible.

Cálculo el valor estimado para cada una de las funciones de hashing para ver cuál es la mejor.

$$h(x) = ax + b \bmod 32.$$

$$\cdot (a=2, b=1):$$

$$h(3) = 7 = 00111$$

$$h(1) = 3 = 00011$$

$$h(4) = 9 = 01001$$

$$h(5) = 11 = 01011$$

$$h(9) = 19 = 10011$$

$$h(2) = 5 = 00101$$

$$h(6) = 13 = 01101$$

El valor de r en este caso es 0 y el valor estimado sería 1.

Cabe recordar que esta función es muy mala porque siempre devuelve números ~~impares~~ impares entonces la cantidad de ceros a derecha siempre es par.

- ($a=3, b=7$):

$$h(3) = 16 = 10000$$

$$h(1) = 10 = 01010$$

$$h(4) = 19 = 10011$$

$$h(5) = 22 = 10110$$

$$h(9) = 2 = 00010$$

$$h(2) = 13 = 01101$$

$$h(6) = 25 = 11001$$

La constante máxima de recorrido en 9 entonces el algoritmo termina un valor de $2^4 = 16$.

- ($a=4, b=0$):

$$h(3) = 12 = 01100$$

Este caso también termina 16

$$h(1) = 4 = 00100$$

como momento de orden 0 del

$$h(4) = 16 = 10000$$

Neces.

$$h(5) = 20 = 10100$$

$$h(9) = 4 = 00100$$

$$h(2) = 8 = 01000$$

$$h(6) = 24 = 11000$$

Vemos que la mejor aproximación que puede hacer Elsgol't - Morin para este Neces es cuando $r=3$ creo que ~~W~~ Wimera B. Ninguno de los ejemplos lo cumple si que lleva uno que ~~no~~ lo cumple.

Sugiero valores $a=0, b=2$. Vemos si esto lo cumplen:

• ($a=0, b=2$):

$$h(3) = 5 = 00,01$$

$$h(1) = 3 = 000,11$$

$$h(4) = 6 = 00,110$$

$$h(5) = 7 = 001,11$$

$$h(8) = 11 = 01,011$$

$$h(2) = 4 = 001,00$$

$$h(6) = 8 = 0,1000$$

En este caso tenemos un valor de $r=3$ entonces el algoritmo estima $2^3=8$ que es la potencia de dos más cercana a 7.

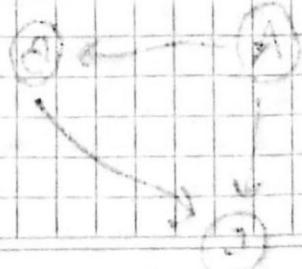
• Sabemos que el resultado final es el resultado de la resta sucesiva de los resultados anteriores.

• La resta sucesiva se hace en el orden de arriba hacia abajo.

• Se hace una resta sucesiva.



• Resta sucesiva, resultado final es 1.



• Resta sucesiva, resultado final es 1.

3 Streaming 10 / 20

✓ - **10 pts** Encuentra un buen valor para a y b pero no generaliza. Es decir no dice que valores de a y b funcionan bien o mal en general.

4 Clustering 0 / 15

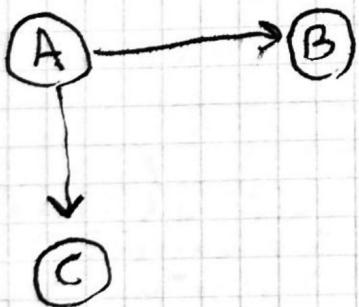
✓ - **15 pts** Sin hacer

5) Asumo que no hay Teletransportación (Lo dijeron en el chat de meet)

En el algoritmo de Page Rank cada nodo tiene un valor de PR que en cada iteración ~~se~~ reporte ~~se~~ igualitativamente entre todos sus vecinos. Como el algoritmo se optimiza hasta la convergencia cognitiva que se llegó a un equilibrio donde cada nodo recibe lo mismo que reporte.

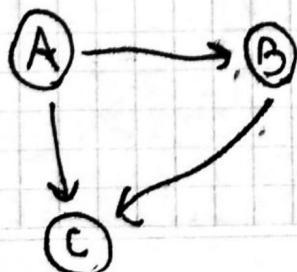
Para encontrar los enlaces de link que salen de B primero vamos a considerar un grafo con los aristas que están si o no:

• A Tiene dos ~~salientes~~ links uno hacia C y otro hacia B.



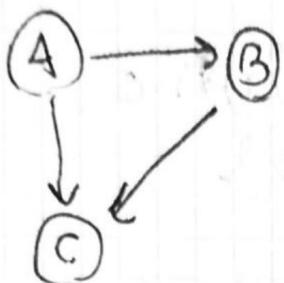
Ahora tenemos que de A no salen más links. (Lo dijeron en el chat de meet).

• C sólo recibe dos links, uno viene de B.



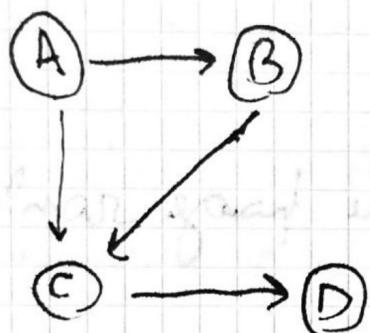
Entonces ahora tenemos que no hay más links hacia C.

- B solo recibe un link.



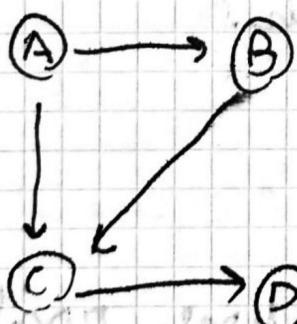
Ahora solo tenemos que ver si hay más links hacia B.

- D solo recibe un link que viene de C.



Ahora solo tenemos que ver si hay más links hacia D.

- Desde C solo vale un link.



Ahora solo tenemos que ver si solo vale uno link de C.

Entendemos parte del grafo armado, ahora voy a unir los dos de los nodos de la parte de page rank.

obviendo que:

- El único link que entra a D es de C
- ~~que el dominio B no tiene enlaces~~
- De C solo recibe un link
- Como D entrega todo su page rank en esta iteración y se llega a una convergencia entonces el valor debe ~~modificarse~~ mantener.

~~Método de la pasividad~~

- D tiene un valor de page rank de 0,1.

Llego a la conclusión:

$$D = I \cdot C$$

valores de ~~pasivo~~ page rank.

$$\Rightarrow C = 0,1$$

Ahora busco el valor de page rank de B obviando que:

- B solo recibe un link y es de A.

- De A solo recibe dos links.
- El valor de page rank de A es 0,15

Llego a la ecuación:

$$B = 0,5A \Rightarrow B = 0,075$$

Observaré si que este valor es constante porque
he llegado a una convergencia.

Con todos los datos obtenidos hasta el momento y sabiendo que:

- C solo recibe dos links, uno viene de A y otro de B.

Llego a la ecuación:

$$C = 0,5A + X B \Rightarrow X = \frac{C - 0,5A}{B} = \frac{1}{3}$$

Donde X es la ~~proporcion~~ porción del page rank de B que ~~se~~ recibe C.

Como B reporte esquitativamente su page rank es $X = \frac{1}{3}$ entonces al B deben venir 3 links.

HAY UNA ACLARACIÓN
EN LA SIG. PÁGINA

Me faltó saber que D no puede ser un
descendiente porque para resolverlos hay que
agregar nuevos ~~o~~ links que vayan a
Sistemas heredados próximos pero hay una
afirmación que dice que B solo recibe un
link y es de A. Esto significa que no
recibe ninguna ~~o~~ de D otra que no es
descendiente.

$$I = \frac{1}{2} \cdot 0.02^2 + \frac{1}{2} \cdot 0.01^2 + \frac{1}{2} \cdot 0.03^2$$

que es la probabilidad de que el sistema
que tiene 2 sistemas heredados sea de tipo A
y que el sistema que tiene 1 sistema heredado
sea de tipo B. La probabilidad de que el sistema
que tiene 3 sistemas heredados sea de tipo C.

0.02, 0.01, 0.03

0.02, 0.01, 0.03

5 Page Rank 15 / 15

✓ - 0 pts Correct

6) Sean recomendador los ítems primero tengo que encontrar los dos más cercanos más cercanos usando la méjorzo- de jaccard:

$$\leftarrow J(A, B) = \frac{1}{6}$$

$$\leftarrow J(A, C) = \frac{2}{4} \Rightarrow \text{Los dos más cercanos son}$$

$$\leftarrow J(A, D) = \frac{1}{5}$$

~~Resalta el efecto de la méjorzo~~

~~WTF~~

Sera ver que ítems recomiendo uno de ítems de votación donde los votos estan ordenados por la méjorzo:

$$\text{Ran: } P_1: \frac{1}{2}, P_3: 0, P_4: \frac{1}{5}, P_7: \frac{1}{3}, P_8: 0$$

~~WTF~~

Se ver que el productor no recomienda von P_1, P_4 y P_7 pero hay un empate entre P_4 y P_7 . Hago colaborative filtering ítem-ítem como se mencionó. Busco los 2 ítems más cercanos a P_4 y a P_7 ~~WTF~~ utilizando la méjorzo- jaccard.

ANEXO

$$\leq J(P_4, P_1) = 0$$

$$\leq J(P_4, P_2) = \frac{1}{2}$$

$$\leq J(P_4, P_3) = 0$$

$$\leq J(P_4, P_5) = 0$$

$$\leq J(P_4, P_6) = 0$$

$$\leq J(P_4, P_7) = \frac{1}{2}$$

$$\leq J(P_4, P_8) = 0$$

Los más cercanos son P_2 y P_7
Así como la votación más favorable
el miembro A.

$$A: 1 \cdot \frac{1}{2} + 0 \cdot \frac{1}{2} = \frac{1}{2}$$

3

$$\leq J(P_7, P_1) = 0$$

$$\leq J(P_7, P_2) = \frac{1}{3}$$

$$\leq J(P_7, P_3) = \frac{1}{2}$$

$$\leq J(P_7, P_4) = \frac{1}{2}$$

$$\leq J(P_7, P_5) = 0$$

$$\leq J(P_7, P_6) = \frac{1}{4}$$

$$\leq J(P_7, P_8) = \frac{1}{2}$$

Como hay 3 con ~~esta~~ menor puntuación $\frac{1}{2}$

Tomo a P_4 y P_9 como más cercanos.

El resultado de la votación

Miér: la votación es favorable

$$A: 0 \cdot \frac{1}{2} + 0 \cdot \frac{1}{2} = 0.$$

(el resultado no habría cambiado
si elegido diferente).

Entonces vemos que P_4 le gana a P_7 .

El orden favor recomendado Miér:

1º: P_1

2º: P_4

3º: P_7 .

6 Recomendaciones 15 / 20

✓ - 5 pts Calcula el item-item contra todos en vez de contra los que ya adquirió A

- 3 No deberia calcular la semejanza contra todos los productos, solo contra los que compró A
- 4 A no compró P3, no deberia ser relevante en el orden de la recomendación

7 Puntos Extra 0 / 15

✓ - **15 pts** Sin puntos adicionales