7506-2020-2 Examen por promocion

Thiago Kovnat

TOTAL POINTS

56 / 100

QUESTION 1

1 Clustering 0 / 20

✓ **- 5 pts** a) mal

✓ **- 5 pts** b) mal.

✓ **- 10 pts** c) mal.

Aprobado con A-

1 Esto está realmente muy mal. Si solo hay 20 inicializaciones posibles no puede haber nunca mas de 20 clusterizaciones finales posibles porque sino desde una inicialización podríamos llegar a clusterizaciones diferentes.

2 Lamentablemente no puedo darte puntos por el a por saber calcular un número combinatorio ya que en el punto b demostrás que no sabés como funciona K-Means. ;(

QUESTION 2

2 Recomendaciones 15 / 20

✓ **- 5 pts** **No muestra los calculos para llegar a los tres usuarios mas similares.**

QUESTION 3

3 PageRank 12 / 20

✓ **- 8 pts** **Errores de cuenta que llegan a resultados imposibles**

3 Evidente error de cuentas el PR total debe sumar 1 y esta muy lejos de esto. Podemos entender errores en una cuenta, pero estos valores son imposibles debio notarlo.

QUESTION 4

4 Streaming 20 / 20

✓ **- 0 pts** **Correct**

QUESTION 5

5 Feature Engineering 9 / 20

Punto a

✓ **- 2 pts** **Elimina datos importantes, como la fecha de publicación o la ciudad.**

✓ **- 3 pts** **Pierde la información de la provincia y de la ciudad.**

✓ **- 1 pts** **Extrae muy poca información del campo Servicios.**

✓ **- 2 pts** **Agrega muy pocos features.**

Punto b

✓ **- 3 pts** **Mal**

4 El estado quizas sea mejor encodearlo con label encoding ya que hay un orden implicito en los datos.

5 Y no agregas columnas para algunos servicios importantes?

6 Solo esto guardas de la ubicación?

7 Se deben normalizar los valores de entrada.

4) La opcion correcta para F3 es la opcion A. La razon es que si uno agarra cada vector y hace la sumatoria de sus componentes esto me da la cantidad de elementos insertados, para F1 y F2 es 20, por lo cual el tercer filtro deberia tener la misma cantidad de elementos. Como A tiene 20, y B tiene 21, la opcion correcta para F3 es A. Para el punto b) la mayor estimacion posible que podemos conseguir es el minimo del maximo de cada filtro. Por lo tanto, el elemento X que tendria esta estimacion seria aquel que hashee a las posiciones donde se encuentra el maximo para cada filtro. Para F1, el maximo es 8 y se encuentra en la posicion 2 (contando desde 0), para F2 el maximo es 6 y se encuentra en la posicion 6 y para F3 el maximo es 5 y se encuentra en la posicion 0. Por lo tanto, la mayor estimacion posible es min(8, 6, 5) = 5. Para obtenerlo el elemento X deberia cumplir h1(x) = 2, h2(x) = 6, h3(x) = 0.

1) a) La cantidad de configuracion iniciales posibles es la cantidad de combinaciones de 3 elementos que se pueden hacer con los elementos dados (Dado que K Means selecciona aleatoriamente los puntos iniciales). Como tengo 6 elementos, la cantidad posible de inicializaciones es la cantidad de combinaciones de 3 elementos que podemos crear con 6 elementos. Utilizando la formula Cm,n = m! / (n! * (m - n)!). En este caso, 6!/(3! * 3!) = 720/36 = 20.

b) La cantidad de clusters posibles hay que calcular la cantidad de permutaciones sin repeticion que se pueden hacer con un set de 6 elementos, lo cual se puede hacer como 6! = 720. En total, hay 720 clusters posibles con los 6 elementos.

c) La maxima cantidad de iteraciones de K Mean se da en el peor caso, es decir, cuando seleccionamos los peores puntos posibles. En este peor caso, la cantidad de iteraciones es 2**(sqrt(n)) siendo en este caso n = 6, por lo que la cantidad maxima de iteraciones es 2**(sqrt(6)) = 6.

1 Clustering 0 / 20

✓ **- 5 pts** a) mal

✓ **- 5 pts** b) mal.

✓ **- 10 pts** c) mal.

Aprobado con A-

1 Esto está realmente muy mal. Si solo hay 20 inicializaciones posibles no puede haber nunca mas de 20 clusterizaciones finales posibles porque sino desde una inicialización podríamos llegar a clusterizaciones diferentes.

2 Lamentablemente no puedo darte puntos por el a por saber calcular un número combinatorio ya que en el punto b demostrás que no sabés como funciona K-Means. ;(

5) Primero, asumimos que los precios ya estan en Dolares, lo cual es comun para las operaciones inmobiliarias.
Lo primero que haria seria crearia una columna que sea si la vivienda es nueva o no utilizando la antiguedad, podria establecer que todas las viviendas nuevas tienen antiguedad 0. Ademas, para poder utilizar el tipo propiedad deberia hacer un encoding ya que esto es una variable categorica la cual no es aceptada por XGBoost, por lo que usaria one-hot-encoding ya que no hay muchos tipos de propiedad posibles, por lo tanto no tendria una cantidad enorme de columnas. Tambien haria una columna booleana la cual indica si la vivienda se encuentra en la capital de la provincia (Como se dice que las propiedades son en el pais implementar esto es relativamente facil) dado que el costo de las viviendas es afectado fuertemente por la locacion, y en general las viviendas en zonas centrales como una capital son mas caras. Deberia tambien hacerle un encoding al estado de la propiedad ya que esta variable afecta el costo de una vivienda. Podria otra vez utilizar OneHotEncoding dada que la cantidad de opciones son limitadas (Como ejemplo tomo Buena, Regular y Mala) Tambien crearia una columna binaria que sea si la propiedad cuenta con los servicios basicos (Luz, agua y Gas). Las columnas dormitorios, cocheras y baños las incluyo sin modificaciones dado que son numericas y su uso es justificado dado que es una medida estandar para una vivienda su cantidad de dormitorios y baños. Por ultimo, agregaria una columna que sea la cantidad de servicios adicionales que tenga la vivienda. Esta columna tendria la cantidad de servicios que no son Luz Agua y Gas (Servicios_Basicos)

Tambien hay que aclarar que previamente habria que hacer una limpieza de datos removiendo los valores nulos del set de datos o reemplazandolos por valores acordes, y pasando las columnas categoricas al tipo categoricas para poder utilizar apropiadamente el One Hot Encoding.

Por lo tanto, las columnas que quedarian para entrenar serian las siguientes: tipo_propiedad_depto, tipo_propiedad_casa, m2_totales, servicios_basicos, propiedad_nueva, casa_capital, estado_buena, estado_mala, estado_regular, dormitorios, cocheras, baños, precio.

Un ejemplo de un registro quedaria como:

TIPO_PROPIEDAD_DEPTO	TIPO_PROPIEDAD_CASA	m2	servicios_basicos	servicios_extra	propiedad_nueva	estado_buena	estado_mala	estado_regular	dormitorios	cocheras	baños	precio
1	0	50	1	2	1	1	0	0	1	1	1	110000

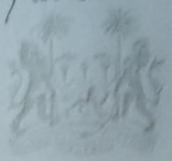
b) En principio, podria utilizar el mismo set trabajado del punto anterior, ya que podria adaptar el modelo de redes neuronales para que utilize los datos que cree en el punto anterior. Sin embargo, al ser un modelo diferente hay probabilidad de que estos datos no se ajusten bien a un modelo de redes neuronales por lo que deberiamos primero observar los resultados utilizando el set de datos del punto anterior y en caso de que los resultados no sean lo que esperabamos cambiar los datos de alguna forma, sea cambiando el encoding, seleccionando features nuevos o removiendo algunos que ya esten presentes. Notemos que las redes neuronales tampoco aceptan features categoricos por lo que el encoding si o si tiene que estar presente.

2) Primero, debemos calcular las semejanza entre el usuario 6 y el resto de los usuarios para quedarnos con los 3 mas similares. Utilizamos la semejanza de Jaccard para esto como lo pide el enunciado y encontramos que los 3 usuarios mas similares son el 2,3 y 4, todos con el mismo valor de 2/3. Observamos a simple vista que dos de esos 3 usuarios indicaron que no les gusto la serie D mientras que a 1 si, como todos tienen la misma semejanza podriamos decir facilmente que si a 1 de los 3 les gusto, la probabilidad de que le guste al usuario 6 es 1/3. Haciendo las cuentas llegamos al mismo resultado: Hacemos la sumatoria de la semejanza multiplicada por el valor que le dio el usuario dividido la sumatoria de las semejanzas. Por lo tanto: R6D = (2/3 * 1 + 2/3 * 0 + 2/3 * 0)/(2/3 + 2/3 + 2/3) = (2/3)/2 = 2/6 = 1/3. Para las cuentas convierto el Si a un 1 y el No a un 0. Entonces, la probabilidad de que al usuario 6 le guste la serie D es 1/3.

2 Recomendaciones 15 / 20

✓ **- 5 pts** **No muestra los calculos para llegar a los tres usuarios mas similares.**

3) Creo la matriz estocastica para el problema, la llamo M



	A	B	C	D	E	F	G	H
A	0	0	0	0	0	1/2	0	0
B	1	0	0	0	1/4	0	0	0
C	0	0	0	1/2	0	0	0	0
D	0	0	0	0	1/4	0	0	0
E	0	1	1	0	0	0	1	0
F	0	0	0	0	1/4	0	0	0
G	0	0	0	0	0	1/2	0	1
H	0	0	0	1/2	1/4	0	0	0

$= M$

Se que para TOPIC-RANK solo debo aplicar teletransportación a paginas de mi topico buscado.

Debo crear un vector de teletransportación apropiada para el problema: Dado que solo me interesan las paginas D, F, H

mi vector seria:

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

$= T$

Tomando $\beta = 0,85$ el problema se reduce a:

$$\beta . M . \underset{\text{inicial}}{\text{VectorPR}} + (1-\beta) . T .$$

Tomo como vector inicial

1/8
1/8
1/8
1/8
1/8
1/8
1/8
1/8

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 17/40 & 0 & 0 \\ \frac{17}{80} & 0 & 0 & 0 & 17/80 & 0 & 0 & 0 \\ 0 & 0 & 0 & 17/20 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{17}{80} & 0 & 0 & 0 \\ 0 & \frac{17}{80} & \frac{17}{80} & 0 & 0 & 0 & \frac{17}{80} & 0 \\ 0 & 0 & 0 & 0 & \frac{17}{80} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{17}{40} & 0 & \frac{17}{20} \\ 0 & 0 & 0 & \frac{17}{40} & \frac{17}{80} & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1/8 \\ 1/8 \\ 1/8 \\ 1/8 \\ 1/8 \\ 1/8 \\ 1/8 \\ 1/8 \end{pmatrix} = \begin{pmatrix} 0.0531 \\ 0.1328 \\ 0.0531 \\ 0.0265 \\ 0.3187 \\ 0.0265 \\ 0.1593 \\ 0.0796 \end{pmatrix} + (1-B) \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1/3 \\ 0 \\ 1/3 \\ 0 \\ 1/3 \end{pmatrix}$$

Tras una iteración: $\begin{pmatrix} 0.0531 \\ 0.1328 \\ 0.0531 \\ 0.0765 \\ 0.3187 \\ 0.0765 \\ 0.1593 \\ 0.1296 \end{pmatrix}$ → LO TOMO COMO VECTOR INICIAL PARA PROX ITER

Segunda iteracion:

$$M.V = \begin{pmatrix} 0.0225 \\ 0.1410 \\ 0.0225 \\ 0.0162 \\ 0.8124 \\ 0.0162 \\ 0.2031 \\ 0.0825 \end{pmatrix} + (1-B) \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1/3 \\ 0 \\ 1/3 \\ 0 \\ 1/3 \end{pmatrix}$$

Tras dos iteraciones el vector queda como:

$$\begin{pmatrix} 0.0225 \\ 0.1410 \\ 0.0225 \\ 0.0662 \\ 0.8124 \\ 0.0662 \\ 0.2031 \\ 0.1325 \end{pmatrix}$$

→ Lo tomo como vector inicial, para la tercera iteración:

3

No llego por tiempo a hacerlo hasta la convergencia pero el proceso es igual: Multiplicar la matriz M por Beta y luego multiplicar eso por el vector de PageRanks hasta el momento. A ese resultado le sumas (1-B) x Vector Teletransportación. El vector resultante se usa como vector de PageRank para la siguiente iteración.

Por lo tanto, el rankeo de las paginas para la segunda iteración es

$$\begin{pmatrix} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{pmatrix} = \begin{pmatrix} 0.0225 \\ 0.1410 \\ 0.0225 \\ 0.0662 \\ 0.8124 \\ 0.0662 \\ 0.2031 \\ 0.1325 \end{pmatrix}$$

Rankeados por PR:

1) E
2) G
3) B
4) H
5) D y F
6) C y A

Habria que elegir un criterio para desempatar. Notar que el empate tiene sentido ya que D y F solo reciben PR de E y a su vez linkean a la misma cantidad de Paginas. El empate entre A y C tambien tiene sentido ya que solo recibe PR de F y D correspondientemente y ambos linkean a una sola pagina.

3 PageRank 12 / 20

✓ **- 8 pts Errores de cuenta que llegan a resultados imposibles**

3 Evidente error de cuentas el PR total debe sumar 1 y esta muy lejos de esto. Podemos entender errores en una cuenta, pero estos valores son imposibles debio notarlo.

4) La opcion correcta para F3 es la opcion A. La razon es que si uno agarra cada vector y hace la sumatoria de sus componentes esto me da la cantidad de elementos insertados, para F1 y F2 es 20, por lo cual el tercer filtro deberia tener la misma cantidad de elementos. Como A tiene 20, y B tiene 21, la opcion correcta para F3 es A. Para el punto b) la mayor estimacion posible que podemos conseguir es el minimo del maximo de cada filtro. Por lo tanto, el elemento X que tendria esta estimacion seria aquel que hashee a las posiciones donde se encuentra el maximo para cada filtro. Para F1, el maximo es 8 y se encuentra en la posicion 2 (contando desde 0), para F2 el maximo es 6 y se encuentra en la posicion 6 y para F3 el maximo es 5 y se encuentra en la posicion 0. Por lo tanto, la mayor estimacion posible es min(8, 6, 5) = 5. Para obtenerlo el elemento X deberia cumplir h1(x) = 2, h2(x) = 6, h3(x) = 0.

1) a) La cantidad de configuracion iniciales posibles es la cantidad de combinaciones de 3 elementos que se pueden hacer con los elementos dados (Dado que K Means selecciona aleatoriamente los puntos iniciales). Como tengo 6 elementos, la cantidad posible de inicializaciones es la cantidad de combinaciones de 3 elementos que podemos crear con 6 elementos. Utilizando la formula Cm,n = m! / (n! * (m - n)!). En este caso, 6!/(3! * 3!) = 720/36 = 20.

b) La cantidad de clusters posibles hay que calcular la cantidad de permutaciones sin repeticion que se pueden hacer con un set de 6 elementos, lo cual se puede hacer como 6! = 720. En total, hay 720 clusters posibles con los 6 elementos.

c) La maxima cantidad de iteraciones de K Mean se da en el peor caso, es decir, cuando seleccionamos los peores puntos posibles. En este peor caso, la cantidad de iteraciones es 2**(sqrt(n)) siendo en este caso n = 6, por lo que la cantidad maxima de iteraciones es 2**(sqrt(6)) = 6.

4 Streaming 20 / 20

✓ - 0 pts Correct

5) Primero, asumimos que los precios ya estan en Dolares, lo cual es comun para las operaciones inmobiliarias.
Lo primero que haria seria crearia una columna que sea si la vivienda es nueva o no utilizando la antiguedad, podria establecer que todas las viviendas nuevas tienen antiguedad 0. Ademas, para poder utilizar el tipo propiedad deberia hacer un encoding ya que esto es una variable categorica la cual no es aceptada por XGBoost, por lo que usaria one-hot-encoding ya que no hay muchos tipos de propiedad posibles, por lo tanto no tendria una cantidad enorme de columnas. Tambien haria una columna booleana la cual indica si la vivienda se encuentra en la capital de la provincia (Como se dice que las propiedades son en el pais implementar esto es relativamente facil) dado que el costo de las viviendas es afectado fuertemente por la locacion, y en general las viviendas en zonas centrales como una capital son mas caras. Deberia tambien hacerle un encoding al estado de la propiedad ya que esta variable afecta el costo de una vivienda. Podria otra vez utilizar OneHotEncoding dada que la cantidad de opciones son limitadas (Como ejemplo tomo Buena, Regular y Mala) Tambien crearia una columna binaria que sea si la propiedad cuenta con los servicios basicos (Luz, agua y Gas). Las columnas dormitorios, cocheras y baños las incluyo sin modificaciones dado que son numericas y su uso es justificado dado que es una medida estandar para una vivienda su cantidad de dormitorios y baños. Por ultimo, agregaria una columna que sea la cantidad de servicios adicionales que tenga la vivienda. Esta columna tendria la cantidad de servicios que no son Luz Agua y Gas (Servicios_Basicos)

Tambien hay que aclarar que previamente habria que hacer una limpieza de datos removiendo los valores nulos del set de datos o reemplazandolos por valores acordes, y pasando las columnas categoricas al tipo categoricas para poder utilizar apropiadamente el One Hot Encoding.

Por lo tanto, las columnas que quedarian para entrenar serian las siguientes: tipo_propiedad_depto, tipo_propiedad_casa, m2_totales, servicios_basicos, propiedad_nueva, casa_capital, estado_buena, estado_mala, estado_regular, dormitorios, cocheras, baños, precio.

Un ejemplo de un registro quedaria como:

TIPO_PROPIEDAD_DEPTO	TIPO_PROPIEDAD_CASA	m2	servicios_basicos	servicios_extra	propiedad_nueva	estado_buena	estado_mala	estado_regular	dormitorios	cocheras	baños	precio
1	0	50	1	2	1	1	0	0	1	1	1	110000

b) En principio, podria utilizar el mismo set trabajado del punto anterior, ya que podria adaptar el modelo de redes neuronales para que utilize los datos que cree en el punto anterior. Sin embargo, al ser un modelo diferente hay probabilidad de que estos datos no se ajusten bien a un modelo de redes neuronales por lo que deberiamos primero observar los resultados utilizando el set de datos del punto anterior y en caso de que los resultados no sean lo que esperabamos cambiar los datos de alguna forma, sea cambiando el encoding, seleccionando features nuevos o removiendo algunos que ya esten presentes. Notemos que las redes neuronales tampoco aceptan features categoricos por lo que el encoding si o si tiene que estar presente.

2) Primero, debemos calcular las semejanza entre el usuario 6 y el resto de los usuarios para quedarnos con los 3 mas similares. Utilizamos la semejanza de Jaccard para esto como lo pide el enunciado y encontramos que los 3 usuarios mas similares son el 2,3 y 4, todos con el mismo valor de 2/3. Observamos a simple vista que dos de esos 3 usuarios indicaron que no les gusto la serie D mientras que a 1 si, como todos tienen la misma semejanza podriamos decir facilmente que si a 1 de los 3 les gusto, la probabilidad de que le guste al usuario 6 es 1/3. Haciendo las cuentas llegamos al mismo resultado: Hacemos la sumatoria de la semejanza multiplicada por el valor que le dio el usuario dividido la sumatoria de las semejanzas. Por lo tanto: R6D = (2/3 * 1 + 2/3 * 0 + 2/3 * 0)/(2/3 + 2/3 + 2/3) = (2/3)/2 = 2/6 = 1/3. Para las cuentas convierto el Si a un 1 y el No a un 0. Entonces, la probabilidad de que al usuario 6 le guste la serie D es 1/3.

5 Feature Engineering 9 / 20

Punto a

✓ **- 2 pts Elimina datos importantes, como la fecha de publicación o la ciudad.**

✓ **- 3 pts Pierde la información de la provincia y de la ciudad.**

✓ **- 1 pts Extrae muy poca información del campo Servicios.**

✓ **- 2 pts Agrega muy pocos features.**

Punto b

✓ **- 3 pts Mal**

4. El estado quizas sea mejor encodearlo con label encoding ya que hay un orden implicito en los datos.
5. Y no agregas columnas para algunos servicios importantes?
6. Solo esto guardas de la ubicación?
7. Se deben normalizar los valores de entrada.