



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Informe del análisis

Sistema de recomendación

Métodos de filtrado colaborativo

C/ Padre Herrera s/n
38207 La Laguna
Santa Cruz de Tenerife. España

T: 900 43 25 26

ull.es



Análisis de los casos

Tenemos 5 tipos de matrices de ejemplo con las que trabajar:

- Matrices 5-10
- Matrices 10-25
- Matrices 25-100
- Matrices 50-250
- Matrices 100-1000

Donde el primer número representa las filas, es decir el número de usuarios que valoran los ítems. Y el segundo representa las columnas, por lo tanto, el número de ítems disponibles para valorar.

En nuestro fichero *"how_execute_with_args.md"* encontraremos diversos ejemplos de cómo se debe ejecutar el programa con sus diversos argumentos especificando todo lo necesario tal y como se muestra en el *"readme.md"*.

Para ayudarnos a realizar un cálculo del tiempo medio de ejecución de una entrada concreta, se ha elaborado el código *"analysis.py"*, que consiste, simplemente, en ejecutar un número determinado de repeticiones nuestro código, midiendo el tiempo transcurrido en cada ejecución, y de esta forma, determinar la media del mismo.

De la misma forma, todos los cálculos, ejemplos y seguimiento se basan en la ejecución real del código con las matrices de ejemplo ubicadas en la carpeta *'examples-utility-matrices'*.

Para los valores, medias y diferencias propuestas a continuación, se ha tenido en cuenta los valores normalizados entre 0 y 1.



Matrices 10-25

```
### INPUT ###
File : examples-utility-matrices/utility-matrix-10-25-1.txt
Metric: pearson
Neighbors: 4
Prediction: media
Use calculated NaN values: True

### OUTPUT ###
-- Utility Matrix --
   Item0  Item1  Item2  ...  Item22  Item23  Item24
User0  0.1634  0.9224  0.8976  ...  0.3142  0.0650  0.2650
User1  0.6616  0.9434  0.2388  ...  0.7932  0.2514  0.5688
User2  0.3642  0.2540  0.7196  ...  0.9200  0.2448  0.9558
User3  0.3300  0.8876  0.7590  ...  0.3010  0.2676  0.0940
User4  0.1490  0.6740  0.6322  ...  NaN  0.4976  0.6862
User5  0.2058  0.7910  0.6760  ...  0.9390  0.1246  0.0186
User6  0.6764  0.9414  0.7966  ...  0.5468  0.3600  0.6116
User7  0.1940  0.8672  0.6800  ...  0.4700  0.2200  0.8074
User8  0.2884  0.0114  0.2752  ...  0.9322  0.5780  0.3922
User9  0.7354  0.2360  0.5988  ...  0.8006  0.3524  0.0668

[10 rows x 25 columns]

-- Similarity --
   User0  User1  User2  ...  User7  User8  User9
User0  #  0.206  0.011  ... -0.005 -0.079 -0.204
User1  0.22  # -0.02  ...  0.105  0.004 -0.311
User2  0.004  0.016  #  ...  0.16  0.331  0.144
User3  0.239  0.044 -0.028  ...  0.317 -0.281  0.114
User4  0.219 -0.11 -0.371  ... -0.079 -0.243 -0.174
User5  0.378  0.074 -0.143  ... -0.021 -0.159  0.237
User6  0.595  0.179 -0.382  ... -0.155 -0.056 -0.075
User7  0.116  0.104  0.16  ...  # -0.072 -0.357
User8  NaN  NaN  NaN  ...  NaN  #  NaN
User9  NaN  NaN  NaN  ...  NaN  NaN  #

[10 rows x 10 columns]

-- Solution Table --
   NaN_Pos  User  Sol_Val  Desn_Val  Neighbors_Selected
0  [2, 22]  User2  0.92  4.60  [8, 9, 1, 0]
1  [1, 4]  User1  0.66  3.30  [0, 6, 7, 3]
2  [1, 21]  User1  0.4  2.00  [0, 6, 7, 3]
3  [3, 0]  User3  0.33  1.65  [7, 0, 9, 1]
4  [3, 8]  User3  0.57  2.85  [7, 0, 9, 1]
...  ...  ...  ...  ...
16  [7, 2]  User7  0.68  3.40  [3, 1, 2, 0]
17  [7, 13]  User7  0.31  1.55  [3, 2, 0, 1]
18  [7, 17]  User7  0.23  1.15  [3, 1, 2, 0]
19  [7, 22]  User7  0.47  2.35  [3, 2, 0, 1]
20  [7, 23]  User7  0.22  1.10  [3, 2, 0, 1]

[21 rows x 5 columns]

-- Incalculable --
[[ 4  9]
 [ 4 22]]
```

Tomando como ejemplo esta ejecución, podemos observar que nuestra matriz de utilidad final se mantiene con dos valores desconocidos. Si observamos el fichero *'tracking.log'*, podremos encontrar que la causa de ello reside en la selección de vecinos. En ambos casos, solo existen tres vecinos del usuario en que sus correlaciones no son negativas, por lo que no cumple nuestra premisa de considerar 4 vecinos. Esto es una tendencia general al usar el método de Pearson. Un gran número de similitudes resultan negativas, lo que suele ser un problema para tomar una muestra de vecinos elevada. Si ejecutamos este ejemplo, añadiendo un vecino más a considerar, es decir, cinco en total, comprobaremos cómo pasamos de 2 valores incalculables a 14.

Ahora bien, si cambiamos la métrica a la de distancia coseno, podremos ver un cambio de tendencia con las similitudes, que en gran medida serán similitudes superiores a 0.6. Si analizamos una muestra de 10 resultados en ambos casos y tomamos una media de la diferencia, veremos que las predicciones varían en torno a un ± 0.105 . Por otro lado, si testeamos con otras matrices de entrada de estas dimensiones, podremos apreciar que en general, el método de distancia coseno es capaz de resolver el problema considerando 7 vecinos sin generar ningún

valor incalculable. Superior a los 4 vecinos observados en Pearson. En cuanto a la distancia euclídea, la media del valor de similitud ronda entre 0.3 y 0.4 de forma muy estable, generando soluciones con una media de diferencia de ± 0.064 con la distancia euclídea y ± 0.078 con el método de pearson.



Matrices 5-10

```
### INPUT ###
File : examples-utility-matrices/utility-matrix-5-10-3.txt
Metric: pearson
Neighbors: 2
Prediction: media
Use calculated NaN values: True

### OUTPUT ###
-- Utility Matrix --
      Item0  Item1  Item2  ...  Item7  Item8  Item9
User0  0.4580  0.1300  0.2614  ...  0.2466  0.3500  0.1146
User1  0.6512  0.4052  0.8102  ...  0.3366  0.5174  NaN
User2  0.5318  0.4414  0.2884  ...  0.7564  0.6924  0.4056
User3  0.4554  0.0752  0.1168  ...  0.8854  0.6080  0.9230
User4  0.9954  0.7750  0.3824  ...  0.7102  0.9496  0.1766

[5 rows x 10 columns]

-- Similarity --
      User0  User1  User2  User3  User4
User0  #  0.659  -0.61  0.382  0.073
User1  0.657  #  -0.796  -0.031  -0.477
User2  NaN  NaN  #  NaN  NaN
User3  NaN  NaN  NaN  #  NaN
User4  0.291  -0.477  0.489  -0.318  #

-- Solution Table --
  NaN_Pos  User  Sol_Val  Desn_Val  Neighbors_Selected
0 [1, 9]  User1  0.37*  1.85  [0]
1 [4, 6]  User4  0.73  3.65  [2, 0]
2 [0, 1]  User0  0.13  0.65  [1, 3]
3 [0, 3]  User0  0.39  1.95  [1, 3]
4 [0, 8]  User0  0.35  1.75  [1, 3]

-- Incalculable --
[[1 9]]
```

En este escenario, empleando las matrices de menor dimensión, vemos que el fenómeno observado en el caso anterior sigue ocurriendo para el método de Pearson, dejando en esta ocasión un valor desconocido incalculable y nuevamente, varios valores de similitud negativos. Con distancia coseno, las similitudes vuelven a ser valores elevados por encima del 0.7 y con una diferencia de predicción con Pearson de ± 0.0375 , y con distancia euclídea las similitudes mantienen en un rango algo más amplio entre 0.4 y 0.6, dejando unas diferencias de predicciones elevadas, ± 0.186 con la distancia coseno y de ± 0.22 con Pearson. Lo que puede reflejar que con matrices pequeñas, la distancia euclídea tiene un comportamiento más distanciado de otras dos métricas.

Si en este escenario estudiamos las diferencias entre el valor de predicción propuesto cuando se cambia el cálculo de predicción de *media* a *simple*, observamos que puede ser un cambio sustancial en los resultados:

	Media de la diferencia
Pearson	± 0.2
Cosine	± 0.286
Euclidean	± 0.168



Matrices 25-100

```
### INPUT ###
File : examples-utility-matrices/utility-matrix-25-100-1.txt
Metric: pearson
Neighbors: 9
Prediction: media
Use calculated NaN values: True

### OUTPUT ###
-- Utility Matrix --
  Item0  Item1  Item2  ...  Item97  Item98  Item99
User0  0.4600  0.1528  0.3474  ...  0.1710  0.7320  0.4144
User1  0.2486  0.2744  0.9496  ...  0.2718  0.8904  0.6852
User2  0.0786  0.6624  0.0548  ...  0.6356  0.4720  0.1514
User3  0.5200  0.1564  0.5682  ...  0.7144  0.7206  0.6690
User4  0.1130  0.7534  0.3022  ...  0.9162  0.3318  0.1356
...
User20 0.3954  0.8998  0.3300  ...  0.0752  0.0186  0.2110
User21 0.3082  0.1982  0.0096  ...  0.7756  0.1508  0.3104
User22 0.3474  0.2474  0.8176  ...  0.6684  0.9332  0.0080
User23 0.4866  0.3032  0.7844  ...  0.6178  0.8718  0.7888
User24 0.5430  0.5508  0.4904  ...  0.9270  0.3462  0.9362

[25 rows x 100 columns]

-- Similarity --
  User0  User1  User2  ...  User22  User23  User24
User0  #  0.102 -0.039  ...  0.12  0.071 -0.064
User1  0.11  # -0.052  ...  0.059 -0.004 -0.179
User2 -0.03 -0.061  #  ...  0.088 -0.146  0.076
User3 -0.055 0.059 -0.014 ... -0.003 -0.194  0.022
User4  NaN  NaN  NaN  ...  NaN  NaN  NaN
...
User20 -0.064 -0.134 0.158 ... 0.006 0.059 -0.17
User21 -0.06 -0.105 0.079 ... -0.039 -0.099 -0.023
User22 0.114 0.059 0.086 ... # 0.234 -0.086
User23 0.078 0.006 -0.133 ... 0.248 # 0.037
User24 -0.06 -0.179 0.075 ... -0.089 0.036 #

[25 rows x 25 columns]

-- Solution Table --
  NaN_Pos  User  Sol_Val  Desn_Val  Neighbors_Selected
0  [14, 31]  User14  0.56  2.80  [0, 10, 6, 7, 16, 19, 15, 5, 12]
1  [21, 50]  User21  0.53  2.65  [11, 9, 18, 7, 12, 2, 4, 6, 13]
2  [3, 0]  User3  0.52  2.60  [17, 6, 19, 12, 1, 24, 7, 8, 20]
3  [11, 4]  User11  0.79  3.95  [18, 8, 21, 15, 10, 12, 19, 7, 9]
4  [24, 80]  User24  0.29  1.45  [5, 18, 16, 12, 2, 8, 23, 3, 13]
...
60 [13, 15]  User13  0.46*  2.30  [12, 7, 9, 1, 21, 6, 24, 14]
61 [13, 64]  User13  0.68*  3.40  [12, 7, 9, 1, 21, 6, 24, 14]
62 [13, 72]  User13  0.27*  1.35  [12, 7, 9, 1, 21, 6, 24, 14]
63 [13, 90]  User13  0.75*  3.75  [12, 7, 9, 1, 21, 6, 24, 14]
64 [13, 98]  User13  0.37*  1.85  [12, 7, 9, 1, 21, 6, 24, 14]

[65 rows x 5 columns]

-- Incalculable --
[[ 7 65]
 [10 42]
 [10 44]
 [10 51]
 [13 15]
 [13 64]
 [13 72]
 [13 90]
 [13 98]]

Complete tracking in the generated tracking.log file
The results have been exported to excel: results.xlsx
Execution time: 1.9076297283172607
```

Una vez más, en este nuevo escenario, las correlaciones negativas de Pearson vuelven a generar valores desconocidos incalculables. Con la distancia coseno, las similitudes vuelven a ser superiores al 0.7 y muy similares entre sí, con una media de diferencia de resultados de +- 0.064. Mientras que con la distancia euclídea las similitudes tienden a ser bajas manteniéndose en un rango entre 0.1 y 0.3, y dejando una diferencia de resultados de +-0.049 con la distancia coseno y 0.026 con Pearson. Como vemos, en este escenario, las diferencias de las predicciones son muy bajas, lo que genera un mayor interés en el uso de la distancia coseno, con la que obtenemos todos los valores desconocidos, similitudes altas y la diferencia de los resultados son muy próximas a los otros métodos, por lo que se perfila como la más completa.

Si volvemos a estudiar la diferencia entre elegir cálculo de predicciones con la media o simple, observamos que el efecto en los resultados es mucho menor que en el caso anterior. En Pearson la media de la diferencia se queda en un +-0.029, en la distancia coseno es de +-0.025 y en la distancia euclidianiana es de +-0.022.



Matrices 50-250

```
### INPUT ###
File : examples-utility-matrices/utility-matrix-50-250-8.txt
Metric: cosine
Neighbors: 18
Prediction: media
Use calculated NaN values: True

### OUTPUT ###
-- Utility Matrix --
      Item0  Item1  Item2  ...  Item247  Item248  Item249
User0  0.5020  0.6862  0.7340  ...  0.1292  0.3210  0.0052
User1  0.7374  0.3114  0.7936  ...  0.6480  0.9474  0.9446
User2  0.4732  0.7366  0.9294  ...  0.9344  0.3006  0.9192
User3  0.8098  0.2736  0.9674  ...  0.4532  0.4198  0.8344
User4  0.3444  0.9334  0.6678  ...  0.6476  0.6208  0.1214
...      ...      ...      ...      ...      ...
User45  0.5800  0.8072  0.2262  ...  0.8670  0.1502  0.5752
User46  0.9000  0.1252  0.4006  ...  0.2294  0.8036  0.1116
User47  0.8228  0.6756  0.3902  ...  0.8642  0.4498  0.2340
User48  0.2242  0.2486  0.8740  ...  0.1848  0.8214  0.4516
User49  0.5166  0.1450  0.9384  ...  0.6230  0.6022  0.1270

[50 rows x 250 columns]

-- Similarity --
      User0  User1  User2  ...  User47  User48  User49
User0      #   NaN   NaN  ...   NaN   NaN   NaN
User1  0.763   #   0.797  ...  0.796  0.763  0.788
User2  0.757  0.796   #   ...  0.778  0.762  0.762
User3  0.745  0.751  0.776  ...  0.775  0.742  0.768
User4  0.742  0.768  0.761  ...  0.806  0.755  0.74
...      ...      ...      ...      ...      ...
User45  0.774  0.753  0.79  ...  0.776  0.751  0.781
User46  0.728  0.744  0.749  ...  0.766  0.777  0.762
User47  0.792  0.797  0.78  ...   #   0.797  0.772
User48  0.752  0.765  0.763  ...  0.8   #   0.8
User49  0.737  0.786  0.763  ...  0.769  0.797   #

[50 rows x 50 columns]

-- Solution Table --
      NaN_Pos  User  Sol_Val  Desn_Val  \
0      [3, 46]  User3    0.51    2.55
1      [19, 180] User19   0.52    2.60
2      [24, 247] User24   0.63    3.15
3      [39, 153] User39   0.44    2.20
4      [4, 27]  User4    0.54    2.70
..      ...      ...      ...      ...
118     [48, 10] User48   0.57    2.85
119     [48, 73] User48   0.60    3.00
120     [48, 132] User48  0.38    1.90
121     [48, 137] User48  0.46    2.30
122     [48, 209] User48  0.57    2.85

Neighbors_Selected
0      [18, 2, 47, 44, 28, 29, 42, 49, 21, 9, 34, 25,...
1      [11, 35, 42, 45, 47, 12, 22, 2, 0, 27, 30, 3, ...
2      [47, 34, 48, 21, 22, 9, 14, 30, 26, 35, 15, 18...
3      [49, 1, 26, 25, 14, 42, 40, 47, 23, 29, 27, 21...
4      [47, 35, 18, 29, 15, 41, 7, 14, 33, 36, 26, 1,...
..      ...
118     [47, 49, 24, 26, 33, 40, 34, 16, 8, 46, 9, 6, ...
119     [47, 49, 24, 26, 33, 40, 34, 8, 16, 46, 9, 6, ...
120     [47, 49, 24, 26, 33, 40, 34, 8, 16, 46, 9, 6, ...
121     [49, 47, 24, 26, 33, 40, 34, 8, 16, 46, 9, 6, ...
122     [47, 49, 24, 26, 33, 40, 34, 8, 16, 46, 9, 6, ...

[123 rows x 5 columns]

-- Incalculable --
[]
```

En este escenario, vamos a analizar el efecto de usar los valores desconocidos calculados para las siguientes operaciones.

Si comparamos las tablas de soluciones en ambos casos, podemos apreciar el efecto que se espera: los primeros valores desconocidos que se calculan, permanecen exactamente iguales en ambas situaciones, sin embargo, si vemos los últimos valores desconocidos, presentan cierta diferencia.

Sin embargo, la diferencia apreciada en este escenario es mínima, siendo para la distancia coseno de +-0.006, para Pearson +-0.001 y para la distancia euclídea +-0.012.

Apreciamos que el efecto repercute mínimamente en Pearson, mientras que en la distancia euclídea parece algo más relevante.



Matrices 100-1000

```
### INPUT ###
File : examples-utility-matrices/utility-matrix-100-1000-1.txt
Metric: pearson
Neighbors: 10
Prediction: media
Use calculated NaN values: True

### OUTPUT ###
-- Utility Matrix --
  Item0  Item1  Item2  ...  Item997  Item998  Item999
User0  0.2454  0.4102  0.7672  ...  0.8964  0.1322  0.1748
User1  0.8826  0.1276  0.5406  ...  0.0852  0.1110  0.7698
User2  0.7050  0.5842  0.1048  ...  0.5920  0.4942  0.9118
User3  0.5084  0.7548  0.0690  ...  0.0894  0.3078  0.8994
User4  0.4248  0.6702  0.8380  ...  0.8922  0.9516  0.1860
...
User95  0.6888  0.8026  0.2940  ...  0.7896  0.4538  0.7094
User96  0.8096  0.1648  0.9692  ...  0.0034  0.1540  0.3056
User97  0.7914  0.2574  0.7572  ...  0.5476  0.5606  0.3554
User98  0.3644  0.3724  0.1242  ...  0.7610  0.7286  0.1930
User99  0.3992  0.0506  0.0822  ...  0.6492  0.4902  0.3942

[100 rows x 1000 columns]

-- Similarity --
  User0  User1  User2  ...  User97  User98  User99
User0  # -0.0 -0.03  ...  0.036  0.019 -0.046
User1 -0.001 # 0.006  ... -0.024  0.008 -0.008
User2  NaN  NaN  #  ...  NaN  NaN  NaN
User3 -0.036 0.057 0.032  ...  0.002 -0.034 -0.012
User4  NaN  NaN  NaN  ...  NaN  NaN  NaN
...
User95 -0.014 -0.01 -0.011  ... -0.035 -0.059 0.013
User96  NaN  NaN  NaN  ...  NaN  NaN  NaN
User97  NaN  NaN  NaN  ...  #  NaN  NaN
User98  0.019 0.008 0.041  ...  0.013  # 0.039
User99 -0.044 -0.008 -0.001  ... -0.023 0.04  #

[100 rows x 100 columns]

-- Solution Table --
  NaN_Pos  User  Sol_Val  Desn_Val  \
0 [16, 698] User16  0.53  2.65
1 [25, 925] User25  0.35  1.75
2 [26, 899] User26  0.39  1.95
3 [36, 962] User36  0.63  3.15
4 [41, 619] User41  0.44  2.20
...
255 [99, 135] User99 0.69  3.45
256 [99, 232] User99 0.55  2.75
257 [99, 505] User99 0.32  1.60
258 [99, 555] User99 0.37  1.85
259 [99, 630] User99 0.50  2.50

Neighbors_Selected
0 [33, 32, 91, 37, 83, 6, 78, 43, 58, 13]
1 [50, 88, 73, 66, 39, 51, 53, 35, 69, 5]
2 [23, 90, 19, 97, 46, 36, 81, 83, 17, 41]
3 [96, 49, 93, 54, 1, 26, 5, 7, 39, 45]
4 [97, 70, 15, 79, 49, 26, 54, 86, 56, 32]
...
255 [90, 15, 77, 33, 6, 46, 34, 69, 12, 39]
256 [90, 15, 77, 33, 6, 46, 69, 34, 39, 98]
257 [90, 15, 77, 33, 6, 46, 69, 34, 39, 98]
258 [90, 15, 77, 33, 6, 46, 69, 34, 39, 98]
259 [90, 15, 77, 33, 46, 6, 69, 34, 39, 98]

[260 rows x 5 columns]

-- Incalculable --
[]

Complete tracking in the generated tracking.log file
The results have been exported to excel: results.xlsx
##### FINAL ANALISIS #####
Tiempo medio de ejecución: 23.6621 segundos
```

Para este escenario, vamos a analizar el tiempo medio de ejecución, haciendo uso de nuestro programa *'analysis.py'* y estableciendo 3 repeticiones. Estudiaremos 3 matrices diferentes, para cada una de ellas, los 3 métodos de cálculo de similitud.

Además, se comparará con otra tabla en que se realiza lo mismo pero para matrices de dimensiones 25-100, pero con 10 repeticiones

Los resultados obtenidos podemos consultarlos en las tablas de las siguientes dos páginas, donde los valores son el tiempo de ejecución medio, medido en segundos.



	10 Vecinos
Matriz 100-1000-1	
Pearson	23.6621
Cosine	28.6286
Euclidean	24.7887
Matriz 100-1000-5	
Pearson	22.6778
Cosine	27.2363
Euclidean	23.7731
Matriz 100-1000-10	
Pearson	21.7069
Cosine	26.7223
Euclidean	23.0491



	10 Vecinos
Matriz 25-100-1	
Pearson	2.3088
Cosine	1.6786
Euclidean	1.5423
Matriz 25-100-5	
Pearson	1.9570
Cosine	1.6859
Euclidean	1.6320
Matriz 25-100-10	
Pearson	1.7766
Cosine	1.6649
Euclidean	1.5525



Conclusiones

De forma general, en todas las ejecuciones de prueba realizadas, se ha podido contemplar un escenario que genera varios valores desconocidos incalculables: un ítem que tenga pocas valoraciones de los usuarios. Como nuestro código tiene un enfoque basado en el usuario, podemos apreciar como la falta de calificaciones de otros usuarios al ítem en cuestión, genera la imposibilidad de realizar una predicción fiable sobre el mismo, al no ser capaces de optar al número de vecinos deseados a contemplar.

Como también hemos visto repetido en múltiples escenarios, el método de Pearson para obtener la correlación entre usuarios, genera un problema por la gran cantidad de similitudes negativas que son rechazadas para la selección de vecinos, por lo qué, si deseamos obtener una matriz de utilidad final con el 100% de los valores desconocidos calculados, tendremos que bajar nuestra exigencia de vecinos a considerar, para tener más opciones de reducir el efecto de estas similitudes negativas. Por otro lado, con el método de la distancia coseno, observamos unas similitudes muy estables generalmente elevadas, por encima del 0.6 en la mayoría de casos. Aunque, estas similitudes tienden a ser muy cercanas entre todas ellas, teniendo que aumentar la precisión a la hora de seleccionar los vecinos con mejor similitud para que esta sea representativa.

En cuánto a la elección del cálculo de predicción entre simple o con media, hemos podido ver que al aumentar las dimensiones de la matriz, la diferencia del valor resuelto entre ambos modos disminuye. Por lo que se presenta como una decisión importante al resultado para matrices de menor tamaño.

Por último, sobre los tiempos de ejecución, en el análisis podemos apreciar el efecto que ocurre entre las matrices de grandes dimensiones y las de una dimensión menor. Con la dimensión 100-1000, Pearson se postula como el método más rápido, mientras que la distancia coseno como el más lento. Sin embargo, en las matrices de menor tamaño, Pearson es el método que más tiempo consume, siendo el que menos la distancia euclídea. Estas diferencias, pueden decantar qué método puede ser el más óptimo según la característica de la matriz de entrada.

Integrantes:

JONATHAN MARTÍNEZ PÉREZ - alu0101254098@ull.edu.es

EDUARDO GONZÁLEZ PÉREZ - alu0101319001@ull.edu.es