Regresión lineal

Amaury Rafael Ortega Camargo

amauryocortega@gmail.com

¿Qué es?

Machine learning

Es un campo donde se usan algoritmos para aprender basados en información para hacer predicciones en el futuro.

- Carros que se manejan solos
- Predicciones de valores en la bolsa

¿Qué usaremos?

Python

- Scikit-learn
- Pandas
- Matplotlib

¿Qué usaremos?

Python

- Scikit-learn
  - Herramientas para minería y análisis de datos
  - Construida sobre NumPy, SciPy, y matplotlib
- Pandas
  - Estructuras de datos y herramientas para análisis de datos
- Matplotlib
  - Graficas

DataSet

BoardGamesGeek (http://www.boardgamegeek.com/)

Sean Beck convirtió la información de los juegos en formato csv.

- `name` – Nombre del juego.
- `playingtime` – Tiempo de juego (Por el creador del juego).
- `minplaytime` – Tiempo mínimo de juego (Por el creador del juego).
- `maxplaytime` – Tiempo máximo de juego (Por el creador del juego).
- `minage` – Edad mínima recomendada para jugar.
- `users_rated` – Numero de usuarios que calificaron el juego.
- `average_rating` – Calificación promedio de los usuarios (1-10).
- `total_weights` – Numero de Weights dados por los usuarios. Es una medida de que tan complicado es el juego.
- `average_weight` – Weight promedio (0-5).

DataSet

```
id,type,name,yearpublished,minplayers,maxplayers,playingtime,minplaytime,maxplaytime,minage,users_rated,average_rating,bayes_average_rating,total_owners,total_traders,total_wanters,total_wishers,total_comments,total_weights,average_weight
12333,boardgame,Twilight Struggle,2005,2,2,180,180,180,13,20113,8.33774,8.22186,26647,372,1219,5865,5347,2562,3.4785
120677,boardgame,Terra Mystica,2012,2,5,150,60,150,12,14383,8.28798,8.14232,16519,132,1586,6277,2526,1423,3.8939
102794,boardgame,Caverna: The Cave Farmers,2013,1,7,210,30,210,12,9262,8.28994,8.06886,12230,99,1476,5600,1700,777,3.7761
25613,boardgame,Through the Ages: A Story of Civilization,2006,2,4,240,240,240,12,13294,8.20407,8.05804,14343,362,1084,5075,3378,1642,4.159
3076,boardgame,Puerto Rico,2002,2,5,150,90,150,12,39883,8.14261,8.04524,44362,795,861,5414,9173,5213,3.2943
31260,boardgame,Agricola,2007,1,5,150,30,150,12,39714,8.11957,8.03847,47522,837,958,6402,9310,5065,3.616
124742,boardgame,Android: Netrunner,2012,2,2,45,45,45,14,15281,8.1676,7.97822,24381,680,627,3244,3202,1260,3.3103
96848,boardgame,Mage Knight Board Game,2011,1,4,150,150,150,14,12697,8.15901,7.96929,18769,367,1116,5427,2861,1409,4.1292
84876,boardgame,The Castles of Burgundy,2011,2,4,90,30,90,12,15461,8.07879,7.95011,20558,215,929,3681,3244,1176,3.0442
72125,boardgame,Eclipse,2011,2,6,200,60,200,14,15709,8.07933,7.93244,17611,273,1108,5581,3188,1486,3.6359
2651,boardgame,Power Grid,2004,2,6,120,120,120,12,34422,7.9888,7.91794,38633,550,1171,6157,7531,3998,3.2911
164153,boardgame,Star Wars: Imperial Assault,2014,2,5,90,90,90,0,3980,8.43944,7.91643,8477,57,701,2970,736,360,3.225
115746,boardgame,War of the Ring (second edition),2012,2,4,150,150,150,13,3870,8.35044,7.88643,6257,71,677,2431,771,288,3.9375
121921,boardgame,Robinson Crusoe: Adventures on the Cursed Island,2012,1,4,180,90,180,14,10539,8.09283,7.88503,15896,217,1379,5821,2109,896,3.6328
35677,boardgame,Le Havre,2008,1,5,200,100,200,12,15774,7.99115,7.88172,16429,205,1343,5149,3458,1450,3.7531
28720,boardgame,Brass,2007,3,4,180,120,180,13,8785,8.03071,7.85824,9171,149,798,2858,2259,1012,3.8646
126163,boardgame,Tzolk'in: The Mayan Calendar,2012,2,4,90,90,90,13,12143,7.98673,7.83148,13958,120,1056,3945,2144,933,3.5595
150376,boardgame,Dead of Winter: A Crossroads Game,2014,2,5,210,45,210,14,9188,8.05776,7.82389,13692,144,1086,4956,1602,608,2.9408
68448,boardgame,7 Wonders,2010,2,7,30,30,30,10,36732,7.87047,7.79413,44982,464,1046,5806,7126,2917,2.3384
18602,boardgame,Caylus,2005,2,5,150,60,150,12,19160,7.89829,7.78071,18885,353,878,4011,4984,2894,3.8252
122515,boardgame,Keyflower,2012,2,6,120,90,120,12,6753,7.98786,7.7478,8599,78,1017,3197,1442,517,3.3056
40834,boardgame,Dominion: Intrigue,2009,2,4,30,30,30,13,19261,7.85479,7.73936,26403,374,461,2281,3005,1121,2.4469
62219,boardgame,Dominant Species,2010,2,6,240,120,240,14,10187,7.89276,7.72445,11003,243,1017,4285,2613,1070,4.0103
28143,boardgame,Race for the Galaxy,2007,2,4,60,30,60,12,28655,7.80281,7.72433,33736,708,761,4597,6807,2922,2.9617
103885,boardgame,Star Wars: X-Wing Miniatures Game,2012,2,4,60,60,60,14,11194,7.93301,7.72151,19899,334,421,1826,2183,911,2.4577
93,boardgame,El Grande,1995,2,5,120,60,120,12,15853,7.83279,7.71949,15556,261,1130,3787,4250,1861,3.0919
146021,boardgame,Eldritch Horror,2013,1,8,240,120,240,14,8388,7.97188,7.71614,13668,147,643,3447,1585,674,3.2834
110327,boardgame,Lords of Waterdeep,2012,2,5,60,60,60,12,19864,7.82181,7.70704,24419,257,995,4706,3898,1493,2.5177
37111,boardgame,Battlestar Galactica,2008,3,6,240,120,240,10,20833,7.82752,7.70269,22735,514,643,4005,4817,1783,3.2013
12493,boardgame,Twilight Imperium (Third Edition),2005,3,6,240,180,240,12,12064,7.88473,7.6917,13771,287,927,4650,3369,1761,4.1942
36218,boardgame,Dominion,2008,2,4,30,30,30,13,44319,7.77298,7.68894,55501,1162,511,4362,9078,4089,2.3766
128882,boardgame,The Resistance: Avalon,2012,5,10,30,30,30,13,9639,7.86147,7.67008,14186,184,305,1674,1711,522,1.8027
42,boardgame,Tigris & Euphrates,1997,2,4,90,90,90,12,17762,7.7634,7.65006,18723,442,929,3778,5092,2364,3.533
9609,boardgame,War of the Ring (first edition),2004,2,4,180,180,180,12,8325,7.85143,7.6382,10675,326,506,2025,2756,1337,3.8362
101721,boardgame,Mage Wars Arena,2012,2,2,90,90,90,13,4727,8.04424,7.62543,8258,360,444,2413,1121,439,3.5763
103343,boardgame,A Game of Thrones: The Board Game (Second Edition),2011,3,6,240,120,240,14,11938,7.8099,7.61503,15356,246,589,3232,1929,892,3.5919
104162,boardgame,Descent: Journeys in the Dark (Second Edition),2012,2,5,120,120,120,14,9702,7.82949,7.61347,15731,409,611,3613,1835,750,3.176
157354,boardgame,Five Tribes,2014,2,4,80,40,80,13,7291,7.84675,7.61009,9970,65,1058,3904,1380,518,2.8764
102680,boardgame,Trajan,2011,2,4,90,90,90,12,6871,7.84106,7.60932,8143,143,778,2618,1457,551,3.6334
132531,boardgame,Roll for the Galaxy,2014,2,5,45,45,45,13,4632,7.92585,7.59572,6551,51,960,3269,912,348,2.6667
521,boardgame,Crokinole,1876,2,4,30,30,30,8,6198,7.80539,7.59099,4250,37,543,2436,1803,470,1.2894
144733,boardgame,Russian Railroads,2013,2,4,120,90,120,12,5383,7.8579,7.58423,6134,86,614,1932,1063,470,3.4043
126042,boardgame,Nations,2013,1,5,200,40,200,14,5546,7.86088,7.5836,6457,145,678,2413,1129,489,3.4949
30549,boardgame,Pandemic,2007,2,4,45,45,45,8,43788,7.65991,7.58194,59298,1088,634,5358,8865,3789,2.4262
123260,boardgame,Suburbia,2012,1,4,90,90,90,8,9775,7.73519,7.57596,11757,151,1180,4085,1908,636,2.7469
73439,boardgame,Troyes,2010,2,4,90,90,90,12,8915,7.73993,7.57297,9259,141,844,2731,1994,784,3.4515
91,boardgame,Paths of Glory,1999,2,2,480,480,480,14,3120,8.0381,7.5726,5554,143,282,1173,1227,529,3.794
70149,boardgame,Ora et Labora,2011,1,4,150,150,150,13,6054,7.81642,7.57172,6995,121,902,2772,1225,516,3.8953
14105,boardgame,Commands & Colors: Ancients,2006,2,2,60,60,60,12,6080,7.82838,7.57126,7770,213,574,2057,2204,943,2.6914
34635,boardgame,Stone Age,2008,2,4,90,60,90,10,23492,7.65109,7.56068,24018,269,1292,5522,5009,2117,2.5083
155426,boardgame,Castles of Mad King Ludwig,2014,1,4,90,90,90,13,4989,7.85323,7.55488,7167,63,1078,3579,996,357,2.6695
21050,boardgame,Combat Commander: Europe,2006,2,2,180,60,180,12,3786,7.94325,7.54767,5321,120,351,1281,1477,602,3.1993
147020,boardgame,Star Realms,2014,2,2,20,20,20,12,8821,7.73391,7.54021,14336,181,317,1663,1787,607,1.9654
124361,boardgame,Concordia,2013,2,5,100,100,100,13,4665,7.84719,7.53827,5079,51,684,1918,1015,376,3.1543
17133,boardgame,Railways of the World,2005,2,6,120,120,120,10,8530,7.71693,7.53663,8688,232,660,2299,2710,1296,3.0247
27833,boardgame,Steam,2009,3,5,90,90,90,10,6827,7.7407,7.53418,7456,176,519,2029,1653,601,3.4493
9216,boardgame,Goa,2004,2,4,90,90,90,12,8663,7.69691,7.53291,8357,181,857,2500,2605,1065,3.3897
161970,boardgame,Alchemists,2014,2,4,120,120,120,14,4065,7.9095,7.52374,6281,71,585,2582,783,349,3.659
43111,boardgame,Chaos in the Old World,2009,3,4,120,60,120,13,9049,7.70053,7.51643,9409,176,644,2742,2137,724,3.1354
93260,boardgame,Summoner Wars: Master Set,2011,2,4,30,30,30,9,6859,7.75639,7.50886,9552,320,440,2093,1364,329,2.5015
4098,boardgame,Age of Steam,2002,1,6,120,120,120,13,6112,7.72862,7.50262,5670,175,401,1475,2098,843,3.9276
555,boardgame,The Princes of Florence,2000,2,5,100,75,100,12,12091,7.63383,7.4979,12640,330,567,1946,3366,1516,3.2559
```

81.313 Lineas

DataSet - Matriz

	1	2	3	4
1	id	type	name	yearpublished
2	12333	boardgame	Twilight Struggle	2005
3	120677	boardgame	Terra Mystica	2012

DataSet - Pandas

Leer e imprimir resúmenes estadísticos del DataSet.

Con la estructura de datos `dataframe`

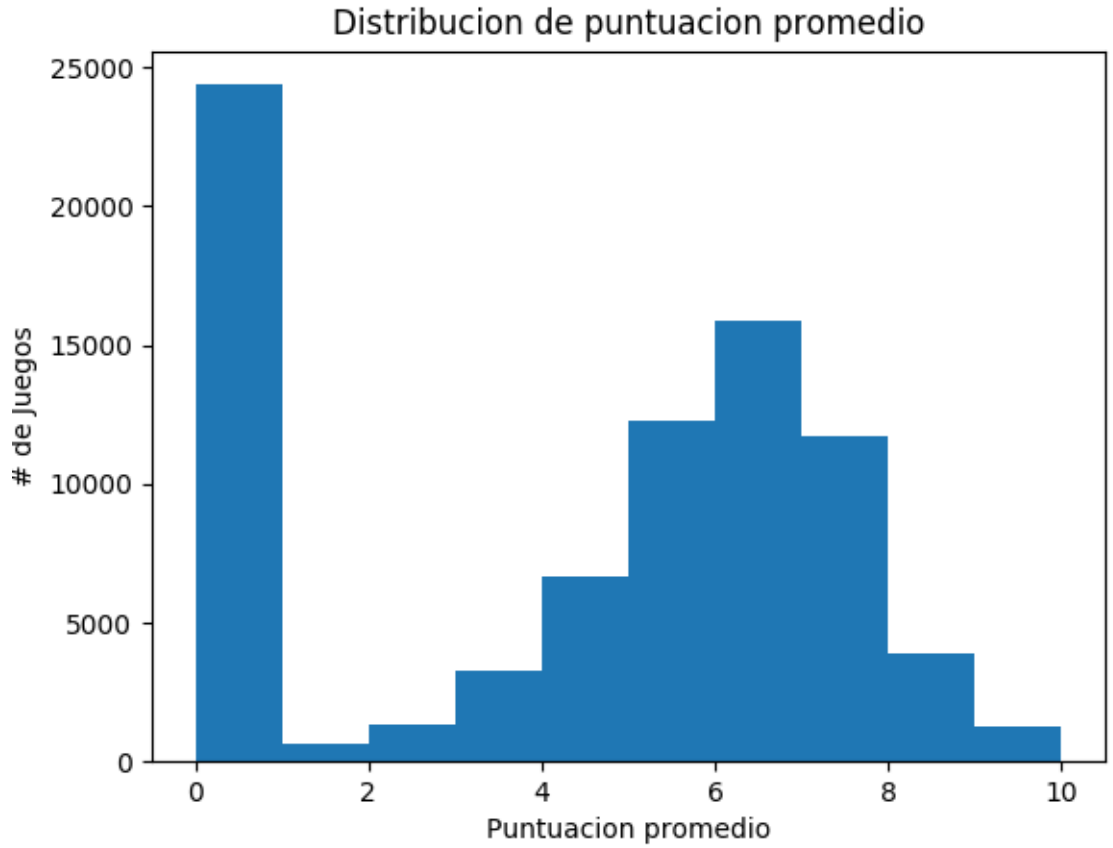
```
import pandas

juegos = pandas.read_csv("games.csv")

print("-"*5 + "Columnas" + "-"*5)
print(juegos.columns)
print()
print("-"*5 + "Tamaño de DataSet (Filas, Columnas)" + "-"*5)
print(juegos.shape)
print()
```

```
-----Columnas-----
Index(['id', 'type', 'name', 'yearpublished', 'minplayers', 'maxplayers',
       'playingtime', 'minplaytime', 'maxplaytime', 'minage', 'users_rated',
       'average_rating', 'bayes_average_rating', 'total_owners',
       'total_traders', 'total_wanters', 'total_wishers', 'total_comments',
       'total_weights', 'average_weight'],
      dtype='object')

-----Tamaño de DataSet (Filas, Columnas)-----
(81312, 20)
```

Distribución de average_rating

```
import matplotlib.pyplot as graficar

 if GRAFICAR:
     graficar.title("Distribucion de puntuacion promedio")
     graficar.xlabel("Puntuacion promedio")
     graficar.ylabel("# de Juegos")
     graficar.hist(juegos["average_rating"])
     graficar.show()
     graficar.clf()
```

Indexación por columna que retorna toda la columna



¿Por qué tantos en 0?

```
print("-"*5 + "Diff entre juego con puntaje de 0 y con puntaje superior a 0" + "-"*5)
print(juegos[juegos["average_rating"] == 0].iloc[0])
print(juegos[juegos["average_rating"] > 0].iloc[0])
print()
```

juegos[juegos["average_rating"] == 0] retornara un dataframe con solo las filas donde el valor de la columna average_rating es 0

iloc[0] retornara toda la primera fila del dataframe

¿Por qué tantos en 0?

```
-----Diff entre juego con puntaje de 0 y con puntaje superior a 0-----
id                          318
type                  boardgame
name                 Looney Leo
yearpublished                 0
minplayers                    0
maxplayers                    0
playingtime                   0
minplaytime                   0
maxplaytime                   0
minage                        0
users_rated                   0
average_rating                0
bayes_average_rating          0
total_owners                  0
total_traders                 0
total_wanters                 0
total_wishers                 1
total_comments                0
total_weights                 0
average_weight                0
Name: 13048, dtype: object
```

```
id                          12333
type                    boardgame
name                Twilight Struggle
yearpublished                2005
minplayers                      2
maxplayers                      2
playingtime                   180
minplaytime                   180
maxplaytime                   180
minage                         13
users_rated                 20113
average_rating            8.33774
bayes_average_rating      8.22186
total_owners                26647
total_traders                 372
total_wanters                1219
total_wishers                5865
total_comments               5347
total_weights                2562
average_weight              3.4785
Name: 0, dtype: object
```

¿Por qué tantos en 0?

```
-----Diff entre juego con puntaje de 0 y con puntaje superior a 0-----
id                              318
type                      boardgame
name                     Looney Leo
yearpublished                     0
minplayers                        0
maxplayers                        0
playingtime                       0
minplaytime                       0
maxplaytime                       0
minage                            0
users_rated                       0
average_rating                    0
bayes_average_rating              0
total_owners                      0
total_traders                     0
total_wanters                     0
total_wishers                     1
total_comments                    0
total_weights                     0
average_weight                    0
Name: 13048, dtype: object
```

```
id                              12333
type                        boardgame
name                 Twilight Struggle
yearpublished                    2005
minplayers                          2
maxplayers                          2
playingtime                       180
minplaytime                       180
maxplaytime                       180
minage                             13
users_rated                     20113
average_rating                8.33774
bayes_average_rating          8.22186
total_owners                    26647
total_traders                     372
total_wanters                    1219
total_wishers                    5865
total_comments                   5347
total_weights                    2562
average_weight                 3.4785
Name: 0, dtype: object
```

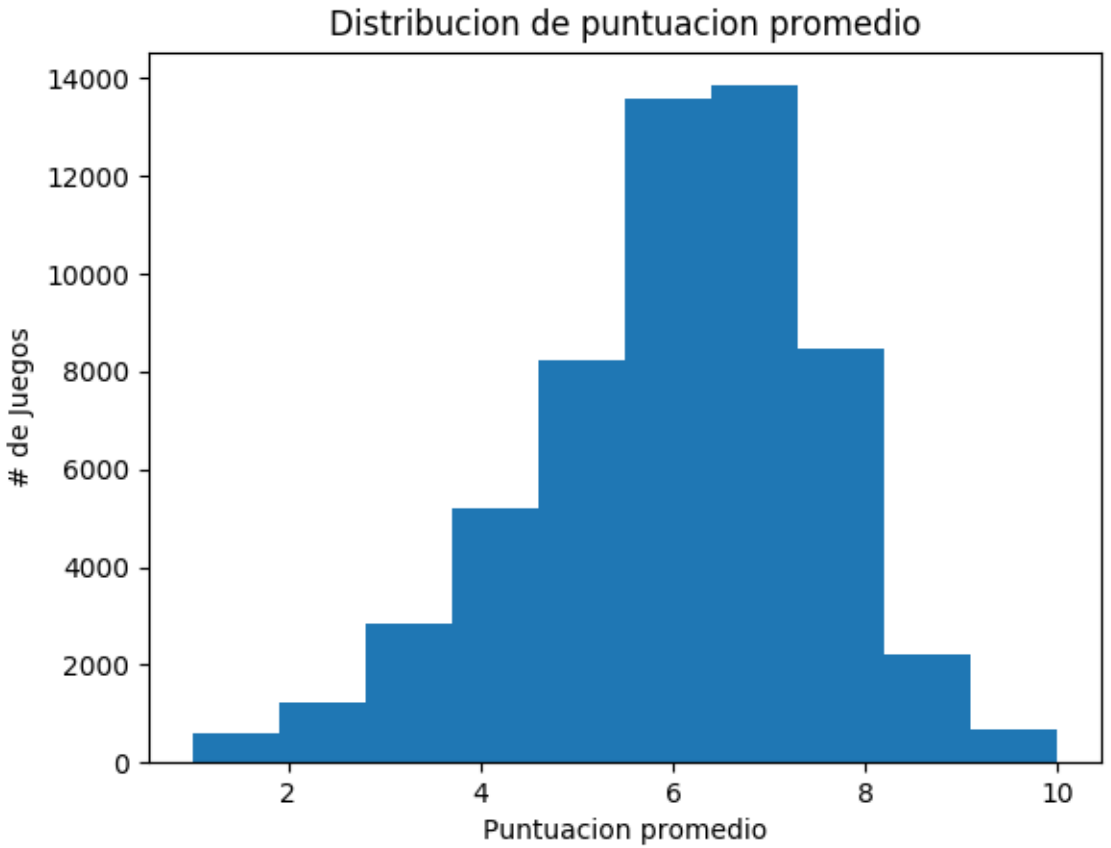
Hay que eliminar el ruido

```
juegos = juegos[juegos["users_rated"] > 0]
juegos = juegos.dropna(axis=0)
```

Remueve cualquier fila que le hagan falta valores

```
if GRAFICAR:
    graficar.title("Distribucion de puntuacion promedio")
    graficar.xlabel("Puntuacion promedio")
    graficar.ylabel("# de Juegos")
    graficar.hist(juegos["average_rating"])
    graficar.show()
    graficar.clf()

print("-"*5 + "Tamaño de DataSet (Filas, Columnas)" + "-"*5)
print(juegos.shape)
print()
```

```
-----Tamaño de DataSet (Filas, Columnas)-----
(56894, 20)
```



Análisis de grupos o agrupamiento

Es la tarea principal de la minería de datos exploratoria y es una técnica común en el análisis de datos estadísticos.

Además es utilizada en múltiples campos como

- Aprendizaje automático
- Reconocimiento de patrones
- Análisis de imágenes
- Búsqueda y recuperación de información
- Bioinformática
- Compresión de datos
- Computación gráfica.

Análisis de grupos o agrupamiento

Un ejemplo de grupo son los juegos que no tienen puntuación.

K-means

Método de agrupamiento, que tiene como objetivo la partición de un conjunto de *n* observaciones en *k* grupos en el que cada observación pertenece al grupo cuyo valor medio es más cercano. Es un método utilizado en minería de datos.

Análisis de grupos o agrupamiento – K-Means

```
from sklearn.cluster import KMeans

modelo_kmeans = KMeans(n_clusters=5, random_state=1)
columnas_numero = juegos._get_numeric_data()
modelo_kmeans.fit(columnas_numero)
etiquetas = modelo_kmeans.labels_
```

Se crea el modelo con 5 clusters y una semilla random de 1

Análisis de grupos o agrupamiento – K-Means

```
from sklearn.cluster import KMeans
modelo_kmeans = KMeans(n_clusters=5, random_state=1)
columnas_numero = juegos._get_numeric_data()
modelo_kmeans.fit(columnas_numero)
etiquetas = modelo_kmeans.labels_

from sklearn.decomposition import PCA as ACP
acp_2 = ACP(2)
columnas_a_graficar = acp_2.fit_transform(columnas_numero)
```

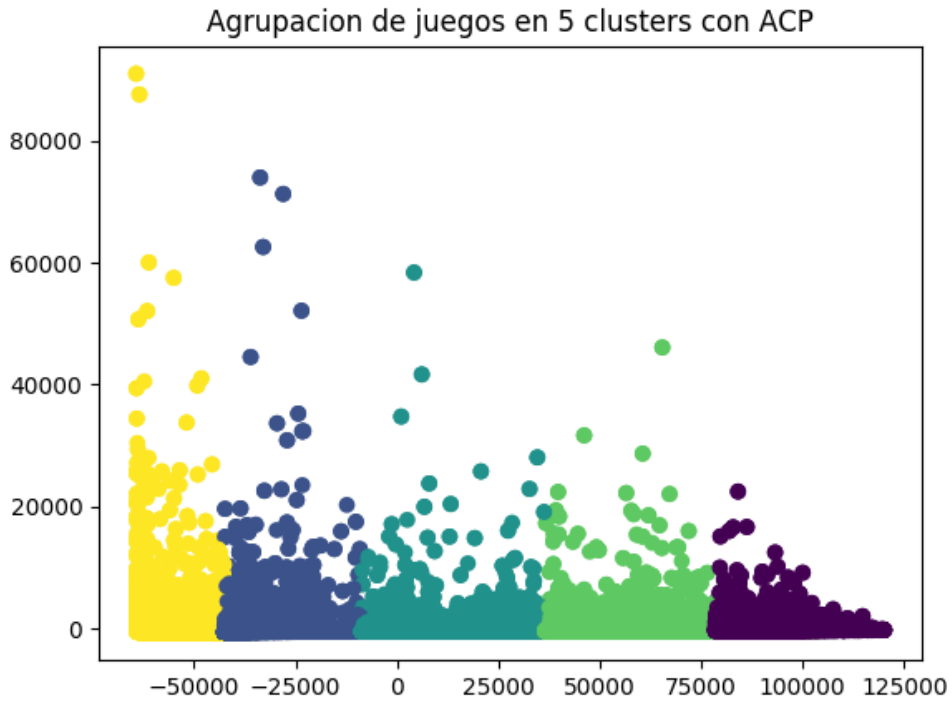
Para visualizar los clusters o grupos, es necesario reducir el numero de columnas debido a que cada columna aumentara el grafico en 1 dimensión.

Análisis de Componentes Principales (En español ACP, en inglés, PCA)

Es una técnica para reducir las dimensiones de un conjunto de datos usando correlación entre columnas.

Análisis de grupos o agrupamiento – K-Means

```
from sklearn.cluster import KMeans
modelo_kmeans = KMeans(n_clusters=5, random_state=1)
columnas_numero = juegos._get_numeric_data()
modelo_kmeans.fit(columnas_numero)
etiquetas = modelo_kmeans.labels_

from sklearn.decomposition import PCA as ACP
acp_2 = ACP(2)
columnas_a_graficar = acp_2.fit_transform(columnas_numero)
```




```
if GRAFICAR:
    graficar.title("Agrupacion de juegos en 5 clusters con ACP")
    graficar.scatter(x=columnas_a_graficar[:,0], y=columnas_a_graficar[:,1], c=etiquetas)
    graficar.show()
    graficar.clf()
```

Inteligencia artificial

Para lograr una predicción correcta, es necesario:

- Medir el error
- Que se va a predecir

Inteligencia artificial

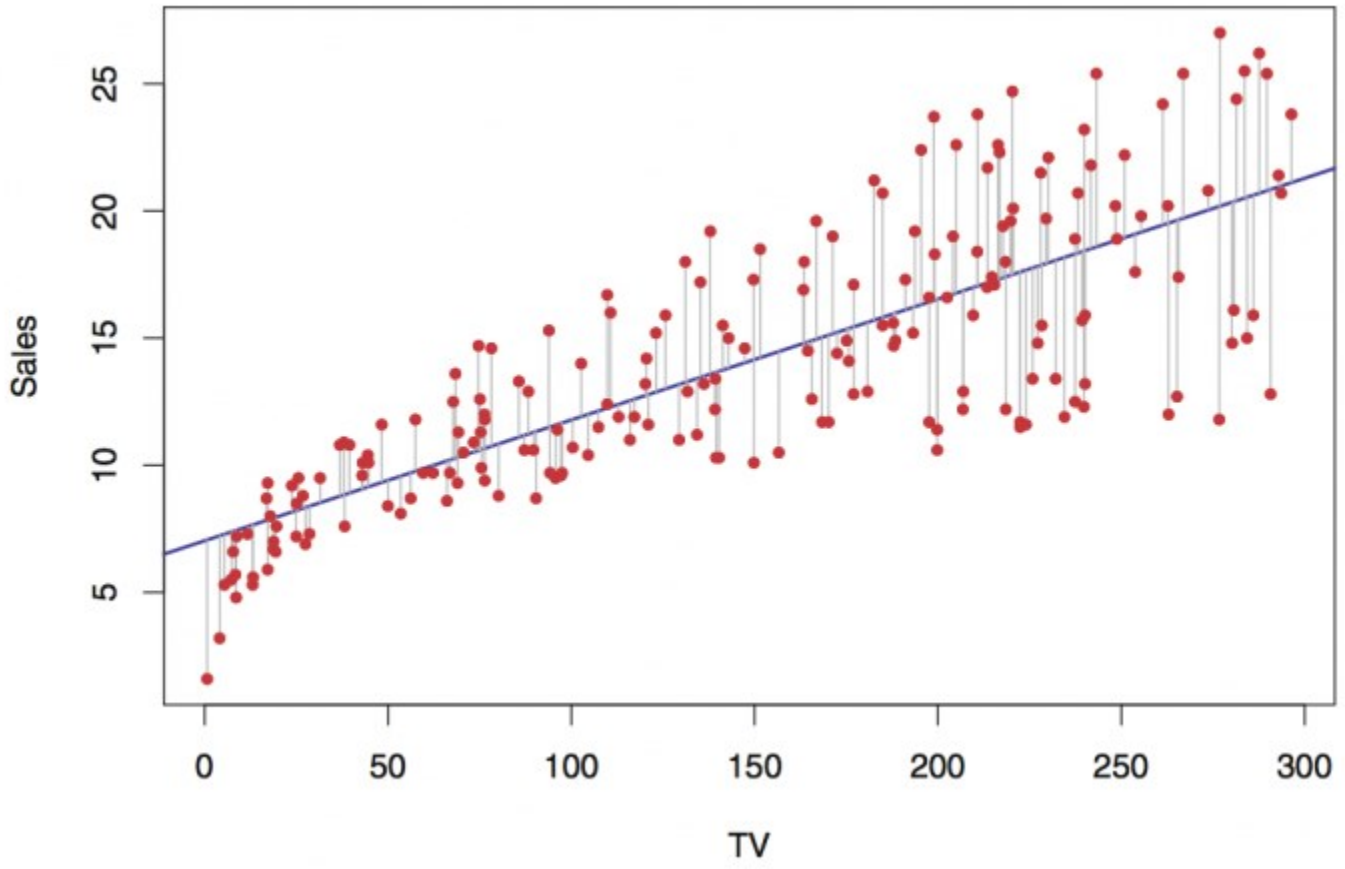
Predecir

El puntaje promedio que se le dará a un juego. (average_rating)

Error - Regresión & variables continuas != Clasificación & variables discretas

Error Cuadrático Medio (En español ECM, en inglés, MSE) porque es rápido de calcular y determina el promedio de que tan distantes están las predicciones de los valores reales

Regresión lineal



En estadística la **regresión lineal** o **ajuste lineal** es un modelo matemático usado para aproximar la relación de dependencia entre una variable dependiente *Y*, las variables independientes X_i y un término aleatorio ε. Este modelo puede ser expresado como:

$$Y_t = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon$$

- Predictores (Columnas)
- Objetivo (Columna a predecir) - average_rating

Correlación

```
print("-"*5 + "Correlacion de average_rating" + "-"*5)
print(juegos.corr()["average_rating"])
print()
```

```
-----Correlacion de average_rating-----
id                      0.304201
yearpublished           0.108461
minplayers             -0.032701
maxplayers             -0.008335
playingtime             0.048994
minplaytime             0.043985
maxplaytime             0.048994
minage                  0.210049
users_rated             0.112564
average_rating          1.000000
bayes_average_rating    0.231563
total_owners            0.137478
total_traders           0.119452
total_wanters           0.196566
total_wishers           0.171375
total_comments          0.123714
total_weights           0.109691
average_weight          0.351081
Name: average_rating, dtype: float64
```

Correlación

```
print("-"*5 + "Correlacion de average_rating" + "-"*5)
print(juegos.corr()["average_rating"])
print()
```

```
-----Correlacion de average_rating-----
id                     0.304201
yearpublished          0.108461
minplayers            -0.032701
maxplayers            -0.008335
playingtime            0.048994
minplaytime            0.043985
maxplaytime            0.048994
minage                 0.210049
users_rated            0.112564
average_rating         1.000000
bayes_average_rating   0.231563
total_owners           0.137478
total_traders          0.119452
total_wanters          0.196566
total_wishers          0.171375
total_comments         0.123714
total_weights          0.109691
average_weight         0.351081
Name: average_rating, dtype: float64
```

- Columnas no numéricas
- Columnas que se calculen usando la columna a predecir average_rating

```
columnas = juegos.columns.tolist()
columnas = [columna for columna in columnas if columna not in ["bayes_average_rating", "average_rating", "type", "name"]]
columna_a_predecir = "average_rating"
```

Regresión lineal – DataSets

Overfitting o sobre-ajuste

Si aprendes 1+1=2 y 2+2=4, serás capaz de responder 1+1 o 2+2 con 0 errores.

Si te preguntan 3+3 no serás capaz de responder.

Como norma, si el algoritmo de aprendizaje produce una cantidad de errores baja es recomendable revisar que no se este presentando un sobre-ajuste.

Regresión lineal – DataSets

```
from sklearn.cross_validation import train_test_split as separar
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error as ECM

set_entrenamiento = juegos.sample(frac=0.8, random_state=1)
set_test = juegos.loc[~juegos.index.isin(set_entrenamiento.index)]

print("-"*5 + "Tamaño de set_entrenamiento (Filas, Columnas)" + "-"*5)
print(set_entrenamiento.shape)
print()
print("-"*5 + "Tamaño de set_test (Filas, Columnas)" + "-"*5)
print(set_test.shape)
print()
```

Se usara el 80% del DataSet para entrenar y el 20% para predecir.

```
-----Tamaño de set_entrenamiento (Filas, Columnas)-----
(45515, 20)

-----Tamaño de set_test (Filas, Columnas)-----
(11379, 20)
```

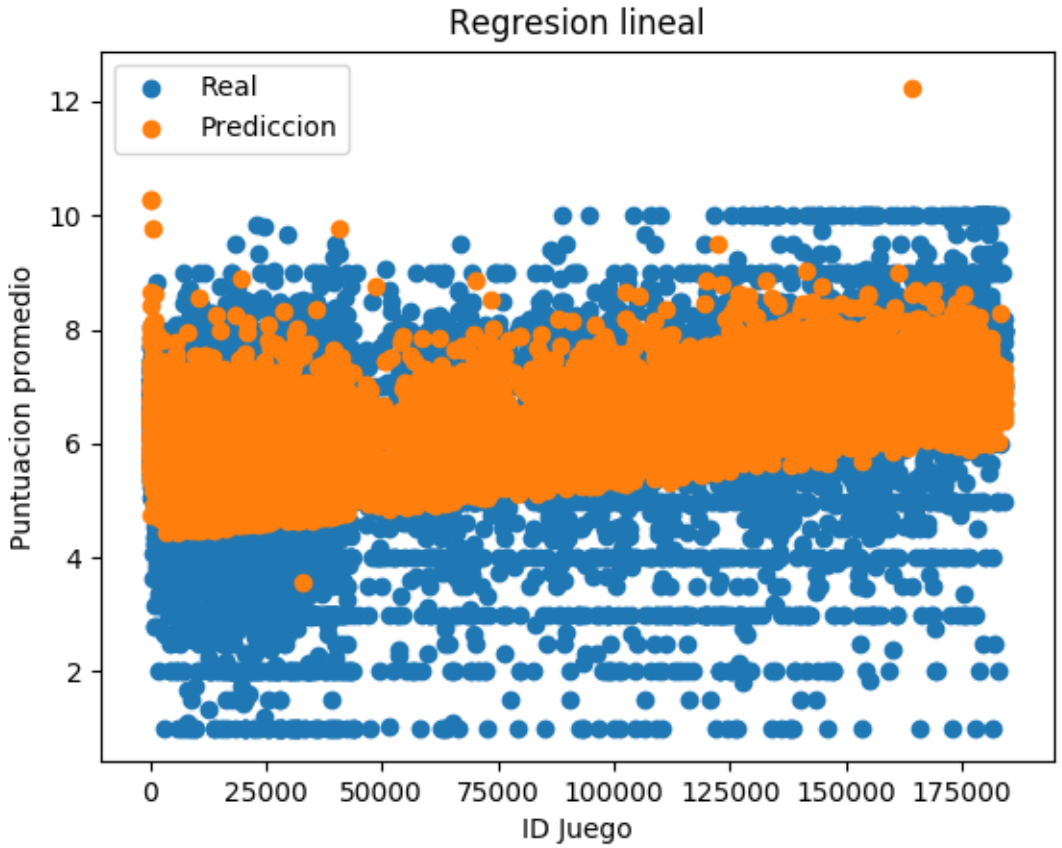
Regresión lineal

```
modelo = LinearRegression()
modelo.fit(set_entrenamiento[columnas], set_entrenamiento[columna_a_predecir])
predicciones = modelo.predict(set_test[columnas])

print("-"*5 + "Predicciones" + "-"*5)
print(predicciones)
print("-"*5 + "VS" + "-"*5)
print(juegos.tail(1)["average_rating"])
print()

print("-"*5 + "Error en prediccion" + "-"*5)
print(ECM(predicciones, set_test[columna_a_predecir]))
print()
```

```
-----Predicciones-----
[ 6.75992275  8.36145643  8.3251177  ...,  6.45530768  6.54520485
  7.1550093 ]
-----VS-----
81279    7.0
Name: average_rating, dtype: float64

-----Error en prediccion-----
1.82392819035
```

Regresión lineal

```
-----Predicciones-----
[ 6.75992275  8.36145643  8.3251177  ...,  6.45530768  6.54520485
  7.1550093 ]
-----VS-----
81279    7.0
Name: average_rating, dtype: float64

-----Error en prediccion-----
1.82392819035
```




```
if GRAFICAR:
    graficar.figure("lineal")
    graficar.title("Regresion lineal")
    graficar.xlabel("ID Juego")
    graficar.ylabel("Puntuacion promedio")
    graficar.scatter(set_test["id"], set_test["average_rating"], label="Real")
    graficar.scatter(set_test["id"], predicciones, label="Prediccion")
    graficar.legend(loc="upper left")
```

Regresión Random Forest

Random forest que es capaz de encontrar correlaciones entre DataSets no lineales cosa que la Regresión lineal no seria capaz

Ej: si minage o edad mínima para un juego afecta a la puntuación

- edad < 5, el puntaje es bajo
- edad 5-10, el puntaje es alto
- edad 10-15, el puntaje es bajo

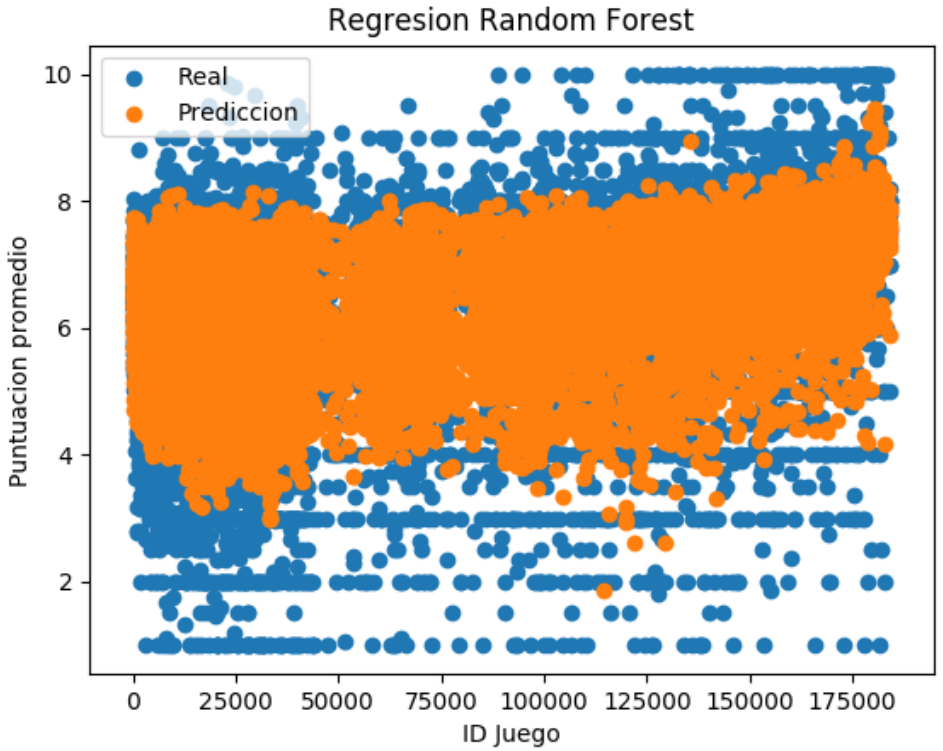
Regresión Random Forest

```
from sklearn.ensemble import RandomForestRegressor

print("-"*5 + "Usando RANDOM FOREST" + "-"*5)
modelo = RandomForestRegressor(n_estimators=100, min_samples_leaf=10, random_state=1)
modelo.fit(set_entrenamiento[columnas], set_entrenamiento[columna_a_predecir])
predicciones = modelo.predict(set_test[columnas])

print("-"*5 + "Predicciones" + "-"*5)
print(predicciones)
print("-"*5 + "VS" + "-"*5)
print(juegos.tail(1)["average_rating"])
print()
print("-"*5 + "Error en prediccion" + "-"*5)
print(ECM(predicciones, set_test[columna_a_predecir]))
print()
```

```
-----Usando RANDOM FOREST-----
-----Predicciones-----
[ 7.87323596  7.76033222  7.87385465 ...,  7.26467783  5.89682253
  7.64492849]
-----VS-----
81279    7.0
Name: average_rating, dtype: float64

-----Error en prediccion-----
1.4144642407
```

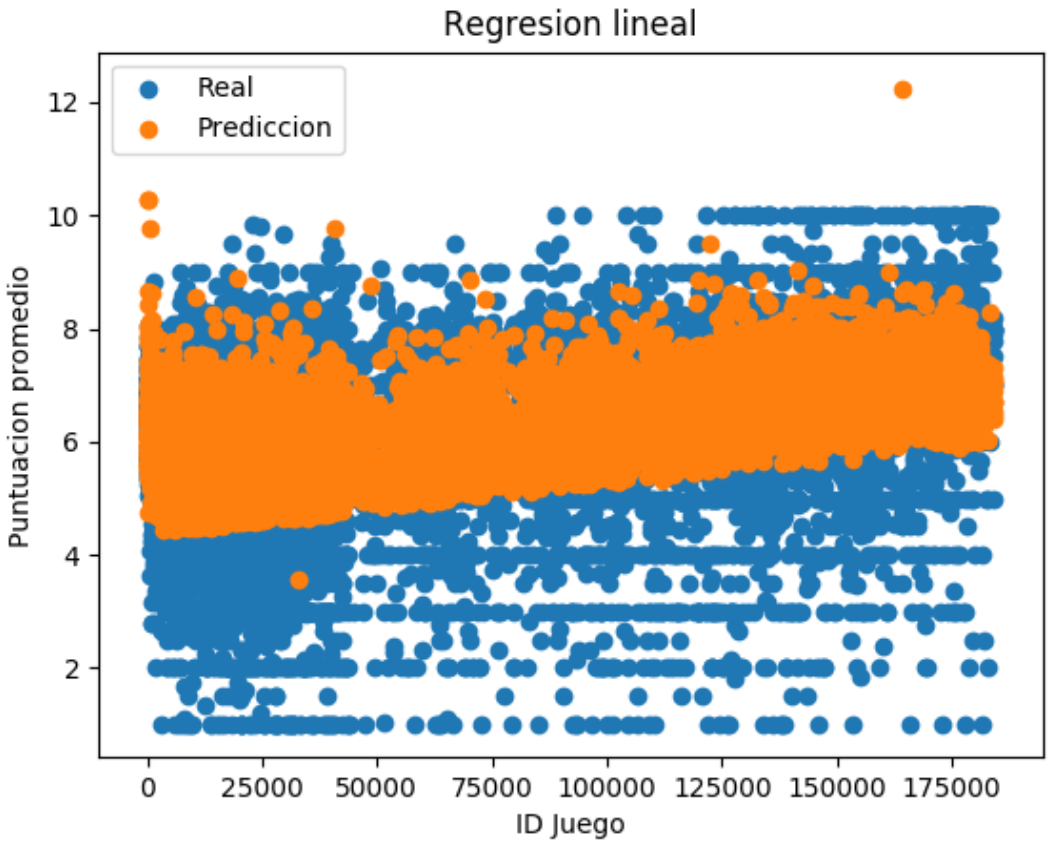
Regresión Random Forest

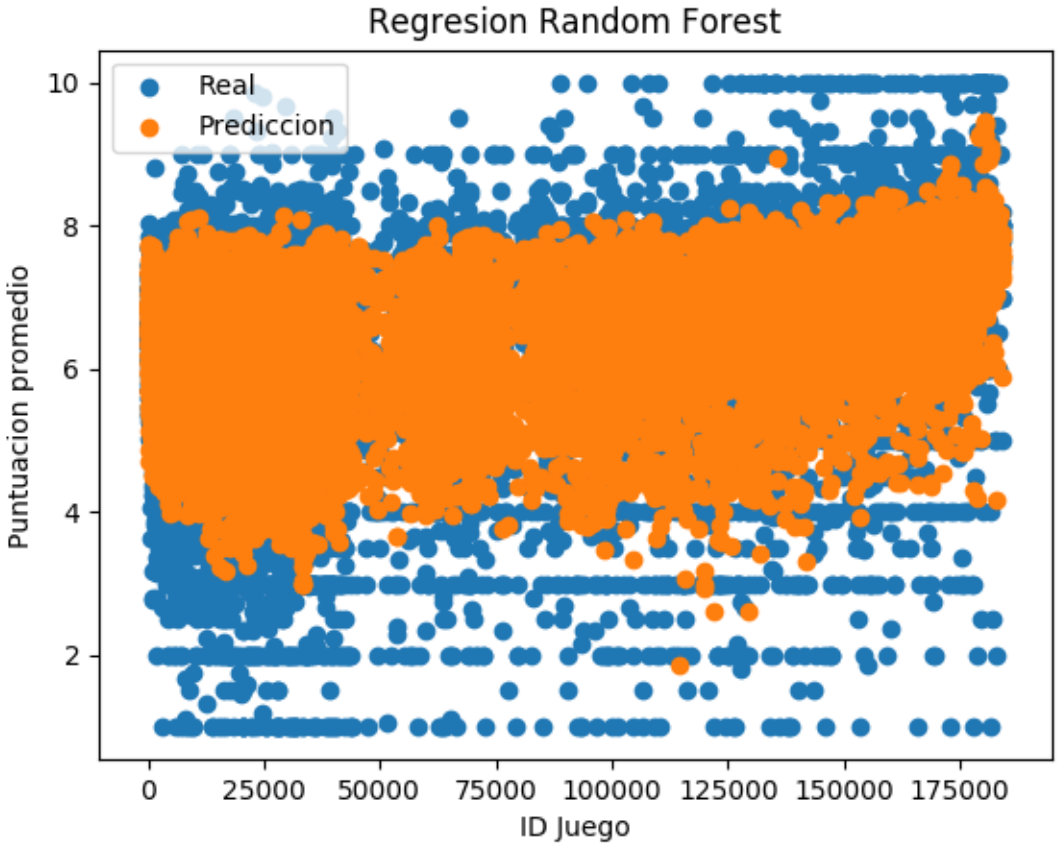
```
-----Usando RANDOM FOREST-----
-----Predicciones-----
[ 7.87323596  7.76033222  7.87385465 ...,  7.26467783  5.89682253
  7.64492849]
-----VS-----
81279    7.0
Name: average_rating, dtype: float64

-----Error en prediccion-----
1.4144642407
```




```
if GRAFICAR:
    graficar.figure("random")
    graficar.title("Regresion Random Forest")
    graficar.xlabel("ID Juego")
    graficar.ylabel("Puntuacion promedio")
    graficar.scatter(set_test["id"], set_test["average_rating"], label="Real")
    graficar.scatter(set_test["id"], predicciones, label="Prediccion")
    graficar.legend(loc="upper left")
    graficar.show()
```

Regresión lineal vs Regresión Random Forest





Codigo

https://github.com/AmauryOrtega/AI-Python

Bibliografía

- Modulo scikit-learn. Machine Learning in Python.
  http://scikit-learn.org/stable/
- Modulo Matplotlib
  https://matplotlib.org/
- Python Data Analysis Library.
  http://pandas.pydata.org/
- Machine Learning Exercises In Python. John Wittenauer.
  http://www.johnwittenauer.net/machine-learning-exercises-in-python-part-2/
- Machine learning with Python: A Tutorial por Vik Paruchuri (2015).
  https://www.dataquest.io/blog/machine-learning-python/
- DataSet por Sean Beck.
  https://github.com/ThaWeatherman
- BoardGamesGeek
  http://www.boardgamegeek.com/