Video Games Sales 2019

Daniel Gutiérrez Rodríguez

1563389

Índice

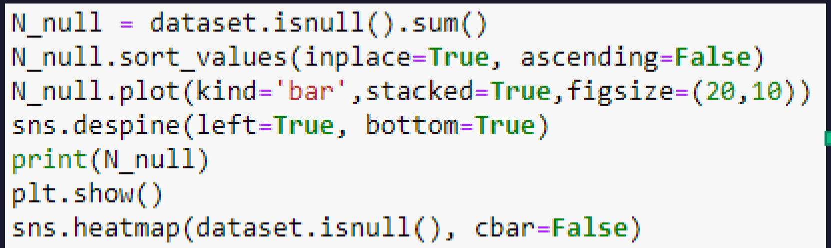
Análisis del dataset

Preparación de los datos

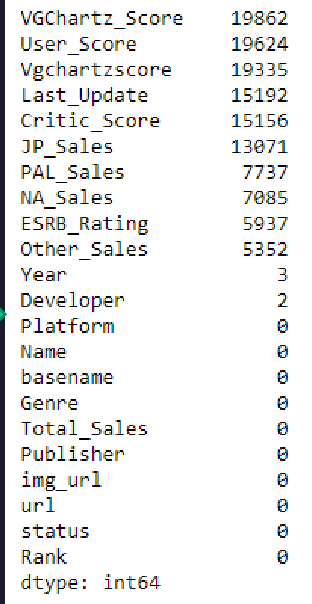
Fase de entrenamiento

Fase de validación

Conclusión



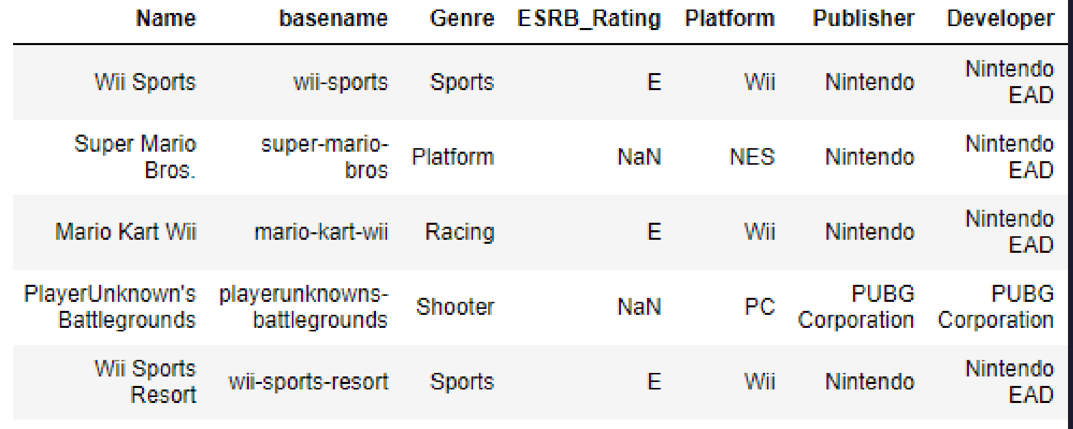
```
N_null = dataset.isnull().sum()
N_null.sort_values(inplace=True, ascending=False)
N_null.plot(kind='bar',stacked=True,figsize=(20,10))
sns.despine(left=True, bottom=True)
print(N_null)
plt.show()
sns.heatmap(dataset.isnull(), cbar=False)
```



```
VGChartz_Score     19862
User_Score         19624
Vgchartzscore      19335
Last_Update        15192
Critic_Score       15156
JP_Sales           13071
PAL_Sales           7737
NA_Sales            7085
ESRB_Rating         5937
Other_Sales         5352
Year                   3
Developer              2
Platform               0
Name                   0
basename               0
Genre                  0
Total_Sales            0
Publisher              0
img_url                0
url                    0
status                 0
Rank                   0
dtype: int64
```

Análisis dataset

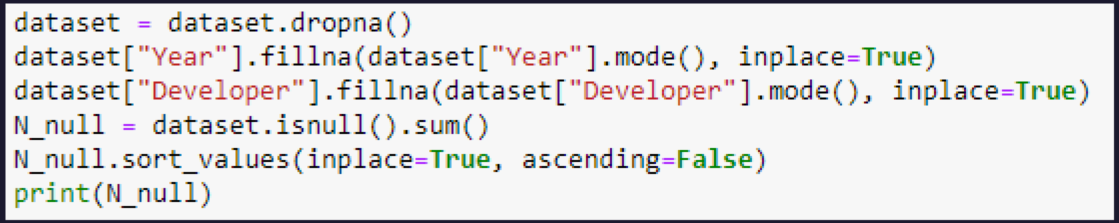
- Multiples valores NaN y Null
- Ventas divididas en 2 columnas



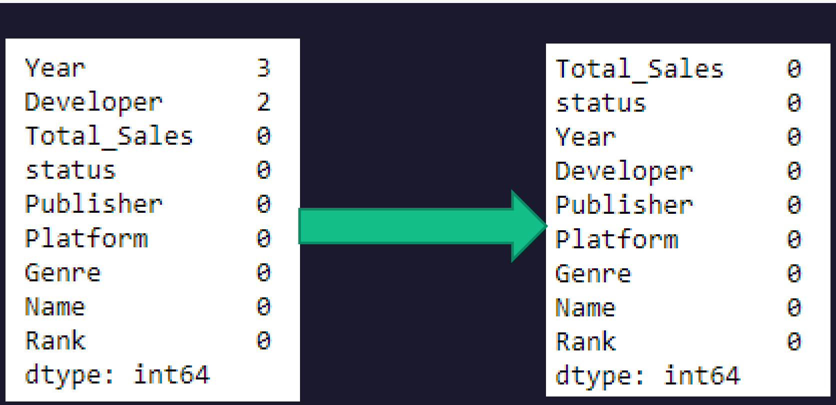
Name	basename	Genre	ESRB_Rating	Platform	Publisher	Developer
Wii Sports	wii-sports	Sports	E	Wii	Nintendo	Nintendo EAD
Super Mario Bros.	super-mario-bros	Platform	NaN	NES	Nintendo	Nintendo EAD
Mario Kart Wii	mario-kart-wii	Racing	E	Wii	Nintendo	Nintendo EAD
PlayerUnknown's Battlegrounds	playerunknowns-battlegrounds	Shooter	NaN	PC	PUBG Corporation	PUBG Corporation
Wii Sports Resort	wii-sports-resort	Sports	E	Wii	Nintendo	Nintendo EAD

Preparación dataset

- Eliminación de las variables
- Normalización de los datos



```
dataset = dataset.dropna()
dataset["Year"].fillna(dataset["Year"].mode(), inplace=True)
dataset["Developer"].fillna(dataset["Developer"].mode(), inplace=True)
N_null = dataset.isnull().sum()
N_null.sort_values(inplace=True, ascending=False)
print(N_null)
```

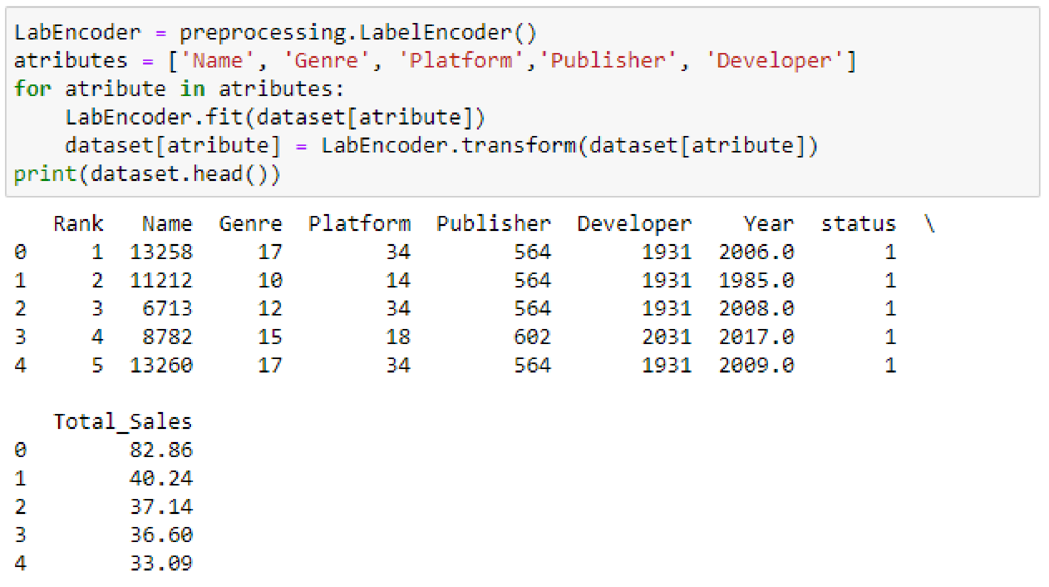


```
Year           3
Developer      2
Total_Sales    0
status         0
Publisher      0
Platform       0
Genre          0
Name           0
Rank           0
dtype: int64
```

```
Total_Sales    0
status         0
Year           0
Developer      0
Publisher      0
Platform       0
Genre          0
Name           0
Rank           0
dtype: int64
```

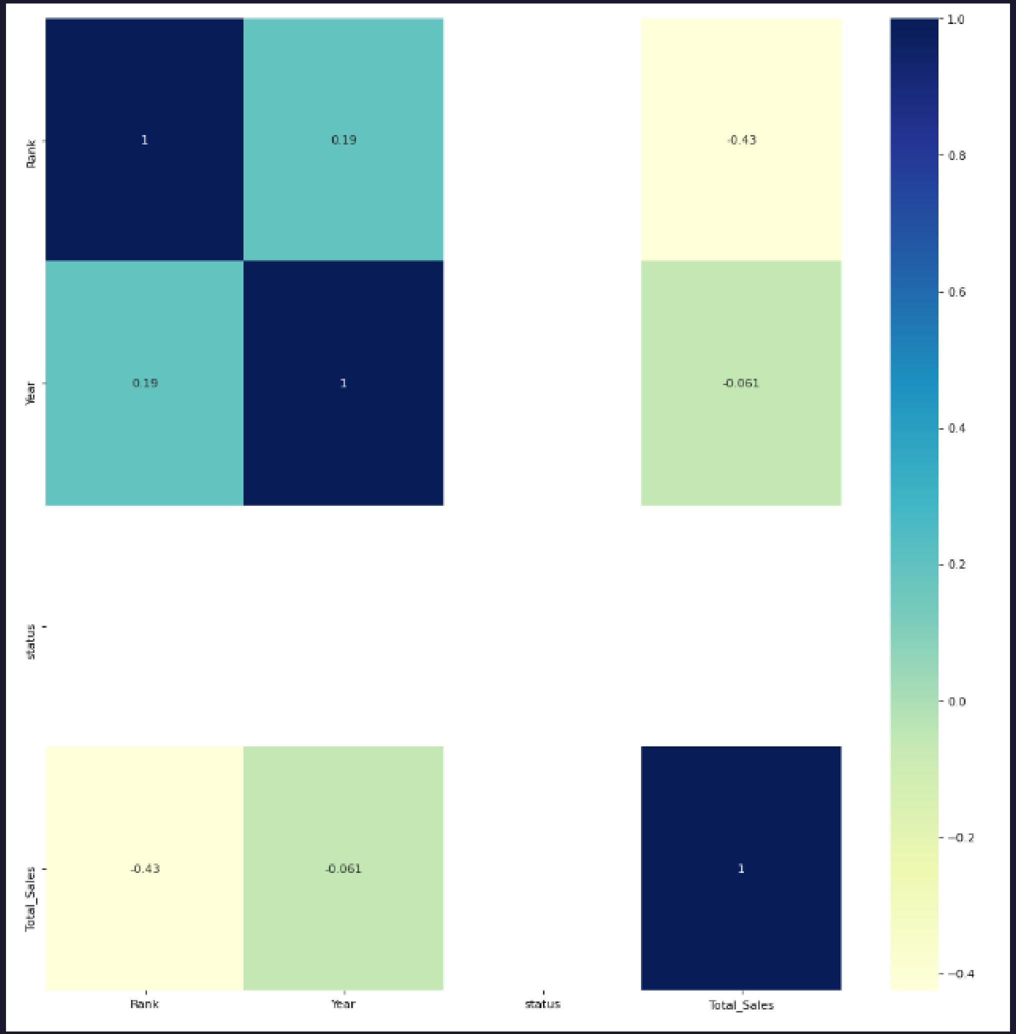
Preparación dataset

- · Falta de algunas variables
- · Analisis variable "Status"

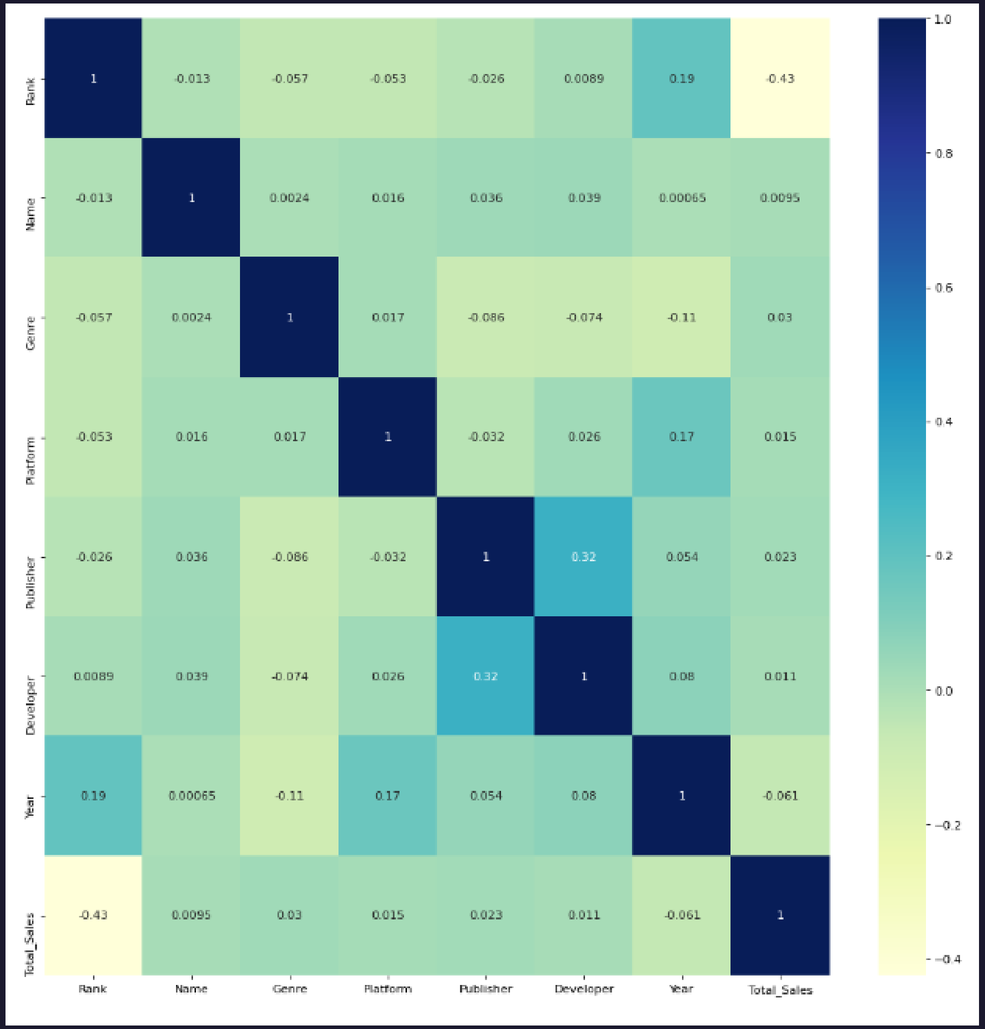


```
LabEncoder = preprocessing.LabelEncoder()
atributes = ['Name', 'Genre', 'Platform','Publisher', 'Developer']
for atribute in atributes:
    LabEncoder.fit(dataset[atribute])
    dataset[atribute] = LabEncoder.transform(dataset[atribute])
print(dataset.head())
```

```
   Rank   Name  Genre  Platform  Publisher  Developer    Year  status  \
0     1  13258     17        34        564       1931  2006.0       1
1     2  11212     10        14        564       1931  1985.0       1
2     3   6713     12        34        564       1931  2008.0       1
3     4   8782     15        18        602       2031  2017.0       1
4     5  13260     17        34        564       1931  2009.0       1

   Total_Sales
0        82.86
1        40.24
2        37.14
3        36.60
4        33.09
```



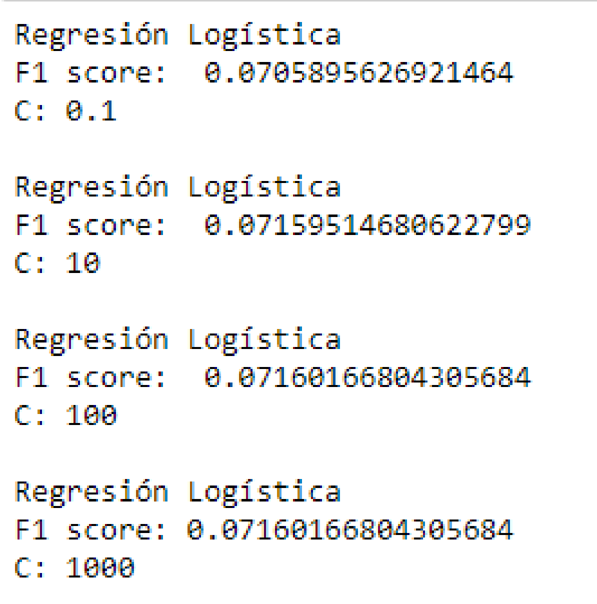
Preparación dataset



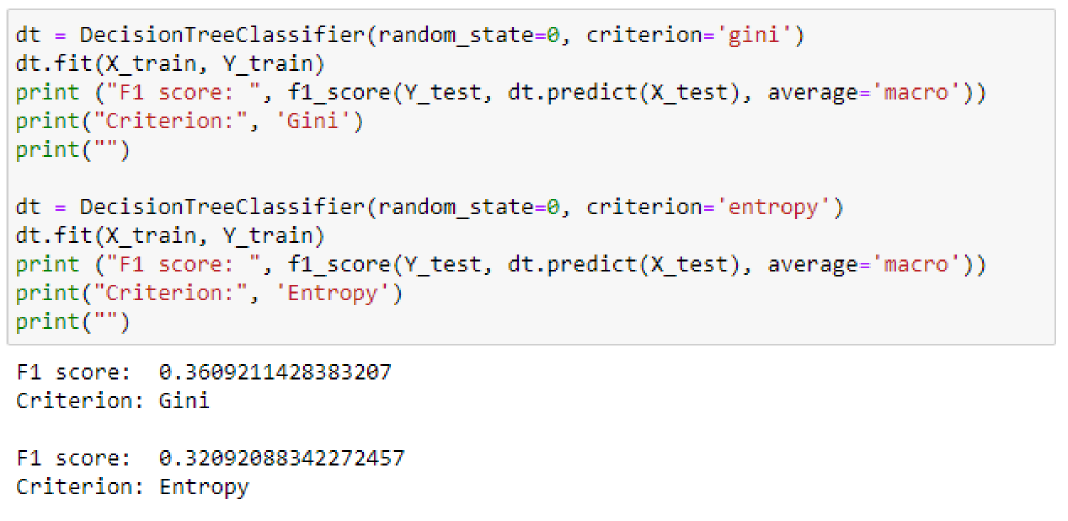
	Rank	Name	Genre	Platform	Publisher	Developer	Year	Total_Sales
Rank	1	-0.013	-0.057	-0.053	-0.026	0.0089	0.19	-0.43
Name	-0.013	1	0.0024	0.016	0.036	0.039	0.00065	0.0095
Genre	-0.057	0.0024	1	0.017	-0.086	-0.074	-0.11	0.03
Platform	-0.053	0.016	0.017	1	-0.032	0.026	0.17	0.015
Publisher	-0.026	0.036	-0.086	-0.032	1	0.32	0.054	0.023
Developer	0.0089	0.039	-0.074	0.026	0.32	1	0.08	0.011
Year	0.19	0.00065	-0.11	0.17	0.054	0.08	1	-0.061
Total_Sales	-0.43	0.0095	0.03	0.015	0.023	0.011	-0.061	1

Entrenamiento

- Decision Tree
- Logistic Regression



```
Regresión Logística
F1 score:  0.0705895626921464
C: 0.1

Regresión Logística
F1 score:  0.07159514680622799
C: 10

Regresión Logística
F1 score:  0.07160166804305684
C: 100

Regresión Logística
F1 score: 0.07160166804305684
C: 1000
```

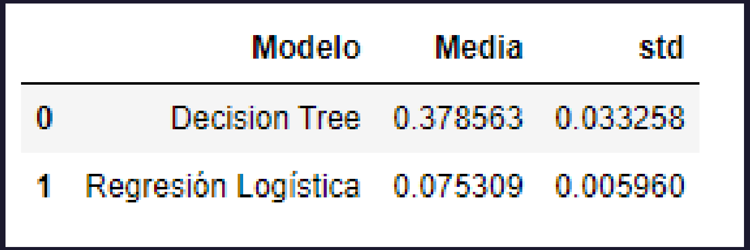


```
dt = DecisionTreeClassifier(random_state=0, criterion='gini')
dt.fit(X_train, Y_train)
print ("F1 score: ", f1_score(Y_test, dt.predict(X_test), average='macro'))
print("Criterion:", 'Gini')
print("")

dt = DecisionTreeClassifier(random_state=0, criterion='entropy')
dt.fit(X_train, Y_train)
print ("F1 score: ", f1_score(Y_test, dt.predict(X_test), average='macro'))
print("Criterion:", 'Entropy')
print("")
```

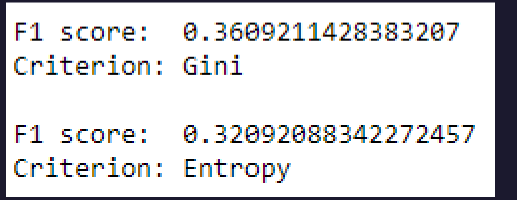
```
F1 score:  0.3609211428383207
Criterion: Gini

F1 score:  0.32092088342272457
Criterion: Entropy
```

Validación

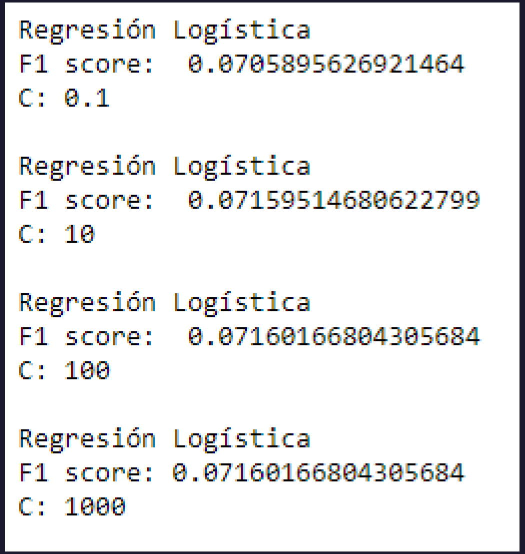
- Decision Tree
- Logistic Regression



	Modelo	Media	std
0	Decision Tree	0.378563	0.033258
1	Regresión Logística	0.075309	0.005960



```
F1 score:  0.3609211428383207
Criterion: Gini

F1 score:  0.32092088342272457
Criterion: Entropy
```



```
Regresión Logística
F1 score:  0.0705895626921464
C: 0.1

Regresión Logística
F1 score:  0.07159514680622799
C: 10

Regresión Logística
F1 score:  0.07160166804305684
C: 100

Regresión Logística
F1 score: 0.07160166804305684
C: 1000
```

Conclusión

Gracias

Daniel Gutiérrez

1563389