COMPETENCIA KAGGLE

Sofía Feilbogen - L. 61889

ÍNDICE

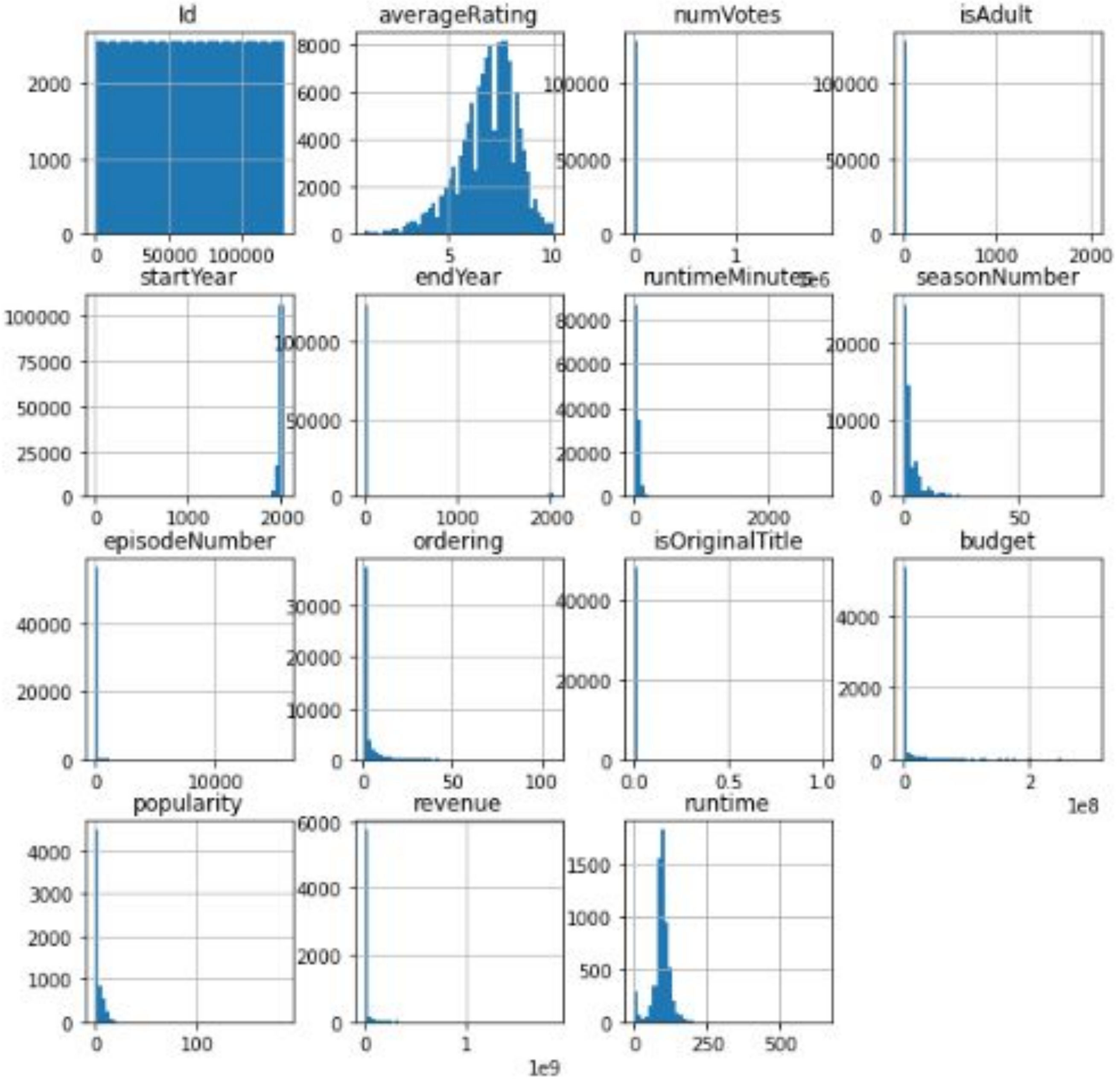
1. Estadísticas descriptivas principales
2. Partición
3. Pipeline
4. Modelos: ajuste de hiperparametros

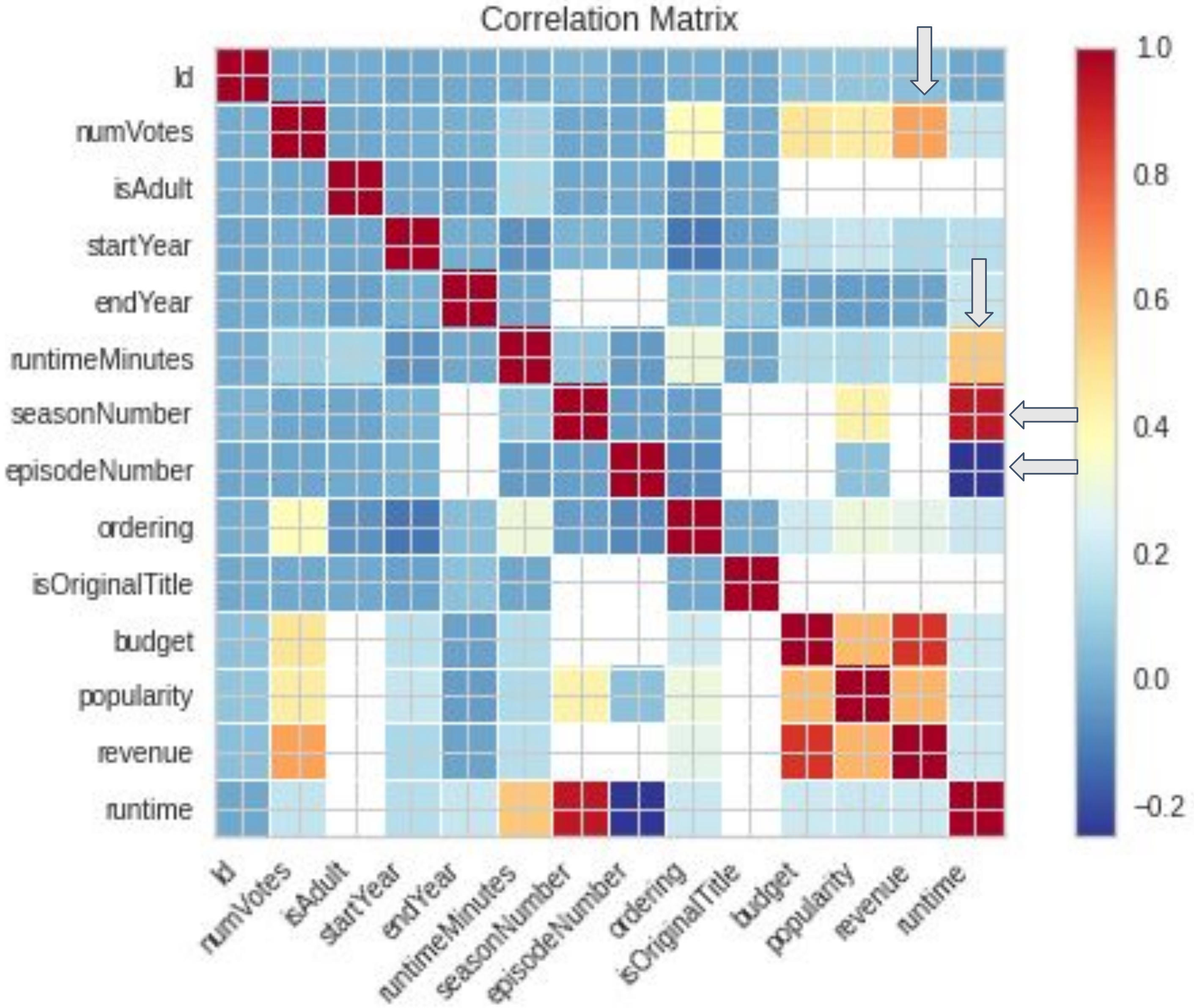
ESTADÍSTICAS DESCRIPTIVAS

Id	averageRating	numVotes	titleType	isAdult	startYear	endYear
runtimeMinutes	genres_x	directors	writers	seasonNumber	episodeNumber	ordering
language	attributes	isOriginalTitle	adult	budget	genres_y	original_language
overview	popularity	production_companies	production_countries	release_date	revenue	runtime
status	tagline	video				

(15178, 31)

	Id	averageRating	numVotes	isAdult	startYear	endYear	runtimeMinutes	seasonNumber	episodeNumber	ordering	isOriginalTitle	budget	popularity	revenue	runtime
count	15178.000000	15178.000000	1.517800e+04	15178.000000	15178.000000	15178.000000	15178.000000	6825.000000	6825.000000	5754.000000	5754.000000	7.180000e+02	718.000000	7.180000e+02	716.000000
mean	7588.500000	6.859013	1.658881e+03	0.020095	1999.148439	62.002042	41.461457	3.983004	57.982857	3.391727	0.000174	5.879068e+06	3.363524	1.742221e+07	93.752793
std	4381.655528	1.418139	2.625960e+04	0.140329	35.605074	346.867339	43.738329	6.061260	628.389557	5.064559	0.013183	2.260087e+07	6.913656	7.823587e+07	30.430507
min	0.000000	1.000000	5.000000e+00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000e+00	0.000000	0.000000e+00	0.000000
25%	3794.250000	6.100000	1.000000e+01	0.000000	1991.000000	0.000000	0.000000	1.000000	3.000000	1.000000	0.000000	0.000000e+00	0.387516	0.000000e+00	86.000000
50%	7588.500000	7.000000	2.300000e+01	0.000000	2007.000000	0.000000	28.000000	2.000000	8.000000	2.000000	0.000000	0.000000e+00	1.081011	0.000000e+00	94.000000
75%	11382.750000	7.800000	9.500000e+01	0.000000	2015.000000	0.000000	75.000000	4.000000	17.000000	3.000000	0.000000	0.000000e+00	4.317293	0.000000e+00	106.250000
max	15177.000000	10.000000	1.493662e+06	1.000000	2021.000000	2021.000000	780.000000	70.000000	14298.000000	106.000000	1.000000	3.000000e+08	133.827820	9.610000e+08	287.000000


Id
averageRating
numVotes
isAdult
startYear
endYear
runtimeMinutes
seasonNumber
episodeNumber
ordering
isOriginalTitle
budget
popularity
revenue
runtime
1e6
1e8
1e9



Correlation Matrix
Id
numVotes
isAdult
startYear
endYear
runtimeMinutes
seasonNumber
episodeNumber
ordering
isOriginalTitle
budget
popularity
revenue
runtime
1.0
0.8
0.6
0.4
0.2
0.0
-0.2

PARTICIÓN

Test size= 0.25
Train size = 0.75

X_train= 11383
y_val=3795

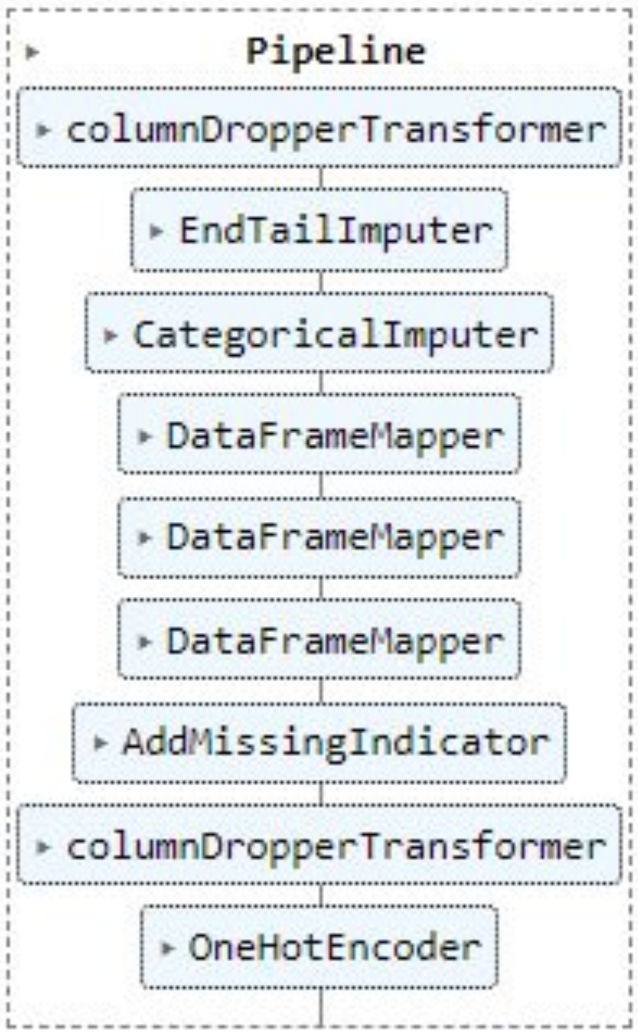
PIPELINE

Fuera del pipeline

1. Agregamos 3 variables para writers, directos y genres_x
2. De las variables categóricas reemplazamos los ceros por NaN

```
[72] X_train["gg"] = X_train["genres_x"].str.split(",")
     X_train["ww"] = X_train["writers"].str.split(",")
     X_train["dd"] = X_train["directors"].str.split(",")
```

```
[73] X_train["titleType"].replace(to_replace = 0, value = np.NaN, inplace=True)
     X_train["genres_x"].replace(to_replace = 0, value = np.NaN, inplace=True)
     X_train["directors"].replace(to_replace = 0, value = np.NaN, inplace=True)
     X_train["writers"].replace(to_replace = 0, value = np.NaN, inplace=True)
     X_train["language"].replace(to_replace = 0, value = np.NaN, inplace=True)
     X_train["attributes"].replace(to_replace = 0, value = np.NaN, inplace=True)
     X_train["adult"].replace(to_replace = 0, value = np.NaN, inplace=True)
     X_train["genres_y"].replace(to_replace = 0, value = np.NaN, inplace=True)
     X_train["original_language"].replace(to_replace = 0, value = np.NaN, inplace=True)
     X_train["overview"].replace(to_replace = 0, value = np.NaN, inplace=True)
     X_train["production_companies"].replace(to_replace = 0, value = np.NaN, inplace=True)
     X_train["production_countries"].replace(to_replace = 0, value = np.NaN, inplace=True)
     X_train["release_date"].replace(to_replace = 0, value = np.NaN, inplace=True)
     X_train["status"].replace(to_replace = 0, value = np.NaN, inplace=True)
     X_train["tagline"].replace(to_replace = 0, value = np.NaN, inplace=True)
     X_train["video"].replace(to_replace = 0, value = np.NaN, inplace=True)
```


Pipeline
columnDropperTransformer
EndTailImputer
CategoricalImputer
DataFrameMapper
DataFrameMapper
DataFrameMapper
AddMissingIndicator
columnDropperTransformer
OneHotEncoder

Dentro del pipeline

1. Eliminamos la variable **Id**
2. De las variables numéricas reemplazamos sus faltantes con **EndTailImputer**.
   - Right tail: mean + 3*std
3. De las variables categóricas reemplazamos sus faltantes categorizandolos como “**Missings**”.
4. Las variables creadas representando writers, directos y genres_x realice un **MultiLabelBinarizer** generando variables dummies por cada una de ellas
5. A las variables de texto que no representan categorías agregue variables **flag** que representan si se encuentra el dato o no: "attributes","overview","tagline"
6. **Elimine** las variables: "attributes","overview","tagline","genres_x","writers","directors","genres_y","production_companies","production_countries"
7. Realice un **OneHotEncoder** con top_categories 3 a las variables categoricas y a “release_date”

```
binarizer1 = DataFrameMapper([
     ("gg", MultiLabelBinarizer(
classes=["Comedy","Drama","Documentary","Action","Missing"])),
], df_out=True, default=None)
```

```
binarizer2 = DataFrameMapper([
     ("dd", MultiLabelBinarizer(
classes=["nm1337210","nm3766090","nm0123273","nm3005544","Missing"])),
], df_out=True, default=None)
```

```
binarizer3 = DataFrameMapper([
     ("ww", MultiLabelBinarizer(
classes=["nm3005544","nm3766090","nm1444457","nm1108327","Missing"])),
], df_out=True, default=None)
```

RandomSampleImputer

Winsorizer

MeanMedianImputer

OutlierTrimmer

```
_=pipe[:5].fit(X_train, y_train)
pipe[:5].transform(X_train)
```

	ww_nm3005544	ww_nm3766090	ww_nm1444457	ww_nm1108327	ww_Missing	attributes_na	overview_na	tagline_na	release_date_Missing	release_date_2005-01-01	release_date_1989-01-01
count	11383.000000	11383.000000	11383.000000	11383.000000	11383.0	11383.000000	11383.000000	11383.000000	11383.000000	11383.000000	11383.000000
mean	0.000703	0.000966	0.001318	0.000615	0.0	0.622683	0.952561	0.974963	0.952385	0.000351	0.000351
std	0.026502	0.031073	0.036279	0.024792	0.0	0.484737	0.212586	0.156245	0.212959	0.018743	0.018743
min	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	1.000000	1.000000	1.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000	0.0	1.000000	1.000000	1.000000	1.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000	0.0	1.000000	1.000000	1.000000	1.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000	1.000000	0.0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

titleType_tvEpisode	titleType_movie	titleType_short	language_Missing	language_0	language_en	adult_Missing	original_language_Missing	original_language_en	original_language_fr
11383.000000	11383.000000	11383.000000	11383.000000	11383.000000	11383.000000	11383.000000	11383.000000	11383.000000	11383.000000
0.448915	0.244136	0.106211	0.622683	0.375999	0.000791	0.999912	0.952297	0.036458	0.002108
0.497405	0.429593	0.308121	0.484737	0.484401	0.028109	0.009373	0.213146	0.187435	0.045871
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000
1.000000	0.000000	0.000000	1.000000	1.000000	0.000000	1.000000	1.000000	0.000000	0.000000
1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

status_Missing	status_Released	status_Planned	video_Missing
11383.000000	11383.000000	11383.000000	11383.000000
0.952297	0.047615	0.000088	0.999912
0.213146	0.212959	0.009373	0.009373
0.000000	0.000000	0.000000	0.000000
1.000000	0.000000	0.000000	1.000000
1.000000	0.000000	0.000000	1.000000
1.000000	0.000000	0.000000	1.000000
1.000000	1.000000	1.000000	1.000000

MODELOS
AJUSTE DE HIPERPARAMETROS

Modelo 1: DecisionTreeRegressor

```
parameters={"splitter":["best","random"],
            "max_depth" : [1,9,12],
            "min_samples_leaf":[1,2,4],
            "max_features":["auto","sqrt",None],

            }

regr_1 = DecisionTreeRegressor(max_depth=4)

M1 = GridSearchCV(regr_1, parameters)
M1.fit(X_train_transformed, y_train)
```

RandomForestRegressor 1

```
[37] parameters = {
            'n_estimators': [100, 300],
            'max_depth': [1,2,4],

     }
     regr_2 = RandomForestRegressor(random_state=0)

     M2 = GridSearchCV(regr_2, parameters)
     M2.fit(X_train_transformed, y_train)
```

RandomForestRegressor 2

```
parameters = {
      'max_depth': [5, None],
      'min_samples_leaf': [1, 4],
      'n_estimators': [100, 200]}

regr_3 = RandomForestRegressor(random_state=0)
M3 = GridSearchCV(regr_3, parameters)
M3.fit(X_train_transformed, y_train)
```

```
parameters4 = {
    'max_depth': [5, None],
    'min_samples_leaf': [1, 4],
    'n_estimators': [100, 200]}

regr_4 = RandomForestRegressor(random_state=0)
M4 = GridSearchCV(regr_4, parameters4)
M4.fit(X_train_transformed, y_train)
```

```
M1=DecisionTreeRegressor(max_depth=12, max_features=None, min_samples_leaf= 4, splitter= 'best')
```

```
M2 = RandomForestRegressor(max_depth= 4,n_estimators=100)
```

```
M3 = RandomForestRegressor(max_depth= 10, min_samples_leaf= 4, n_estimators= 200)
```

```
M4 = RandomForestRegressor(max_depth= None, min_samples_leaf= 4, n_estimators= 200)
```

```
print(M1.score(X_val_transformed,y_val))
```

```
0.14226272429668185
```

```
print(M2.score(X_val_transformed,y_val))
```

```
0.23425140059565142
```

```
print(M3.score(X_val_transformed,y_val))
```

```
0.282461350794433
```

```
print(M4.score(X_val_transformed,y_val))
```

```
0.27393526912314803
```

```
print(model4.score(X_val_transformed,y_val))
```

```
0.3802714857497179
```

```
print(modelo5.score(X_val_transformed,y_val))
```

```
0.29771281893981194
```

Instituto Tecnológico
de Buenos Aires

Muchas gracias