TP2 Análisis Predictivo

Martín Pimentel

Modelo baseline

Features numéricos:

- numVotes
- runtimeMinutes
- isAdult
- age (now().year - startYear)
- seasonNumber
- episodeNumber
- ordering
- isOriginalTitle

Features categóricos:

- genres_x
- titleType
- language

Model, Score:
DecisionTreeRegressor, 0.23

Model, Score:
XGBoost, 0.28

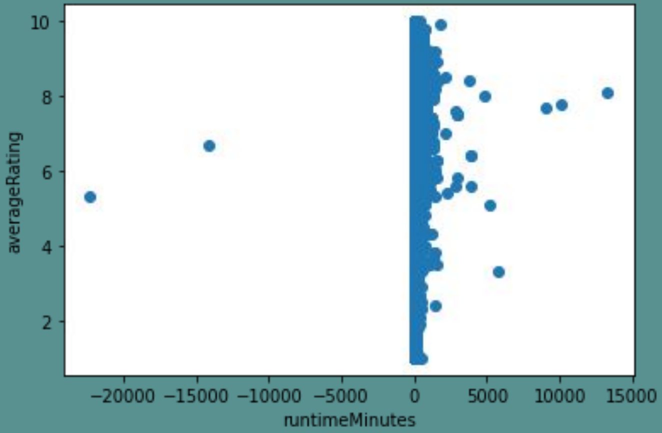
Selección de modelos

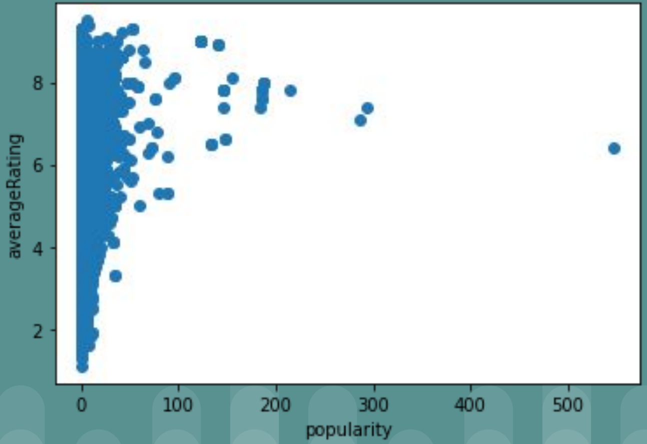
Problemas del dataset:

- Outliers
- N/As

Modelos probados:

- Random Forest
- XGBoost (múltiples veces)
- PCA + XGBoost





Modelo final

XGBoost → 0.421

Features

- numVotes
- runtimeMinutes
- age
- episodeNumber
- seasonNumber
- hasOrdering
- hasPopularity
- hasAttributes
- titleType (encoded)
- genres_x (split & encoded)
- containsTopDirector
- containsTopWriter

Hiperparámetros

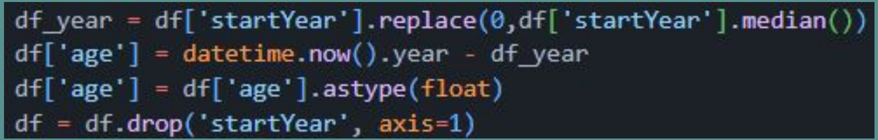
- max_depth = 9
- n_estimators = 700
- gamma = 0 (default)
- lambda = 1 (default)
- learning_rate = 0.25

El resto los dejé en default sin testear.

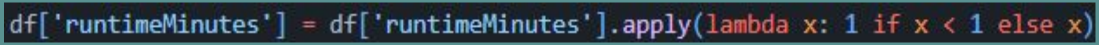
Modelo final

Features

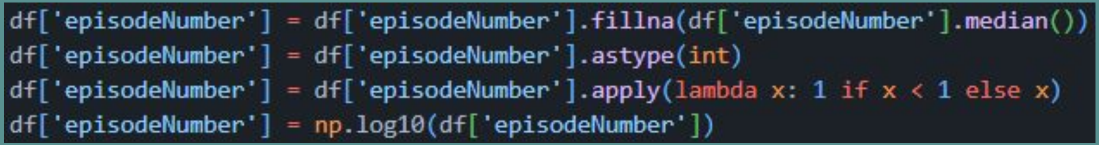
- numVotes
- runtimeMinutes
- age
- episodeNumber
- seasonNumber
- hasOrdering
- hasPopularity
- hasAttributes
- titleType (encoded)
- genres_x (split & encoded)
- containsTopDirector (quantile = 0.9 & appearances > 10)
- containsTopWriter (quantile = 0.9 & appearances > 10)



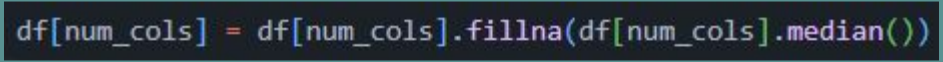
```
df_year = df['startYear'].replace(0,df['startYear'].median())
df['age'] = datetime.now().year - df_year
df['age'] = df['age'].astype(float)
df = df.drop('startYear', axis=1)
```



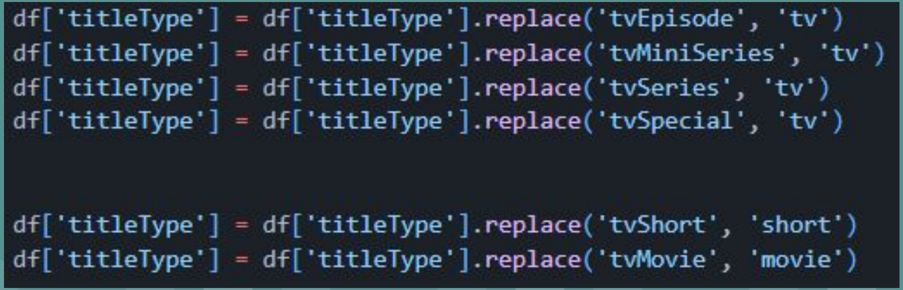
```
df['runtimeMinutes'] = df['runtimeMinutes'].apply(lambda x: 1 if x < 1 else x)
```



```
df['episodeNumber'] = df['episodeNumber'].fillna(df['episodeNumber'].median())
df['episodeNumber'] = df['episodeNumber'].astype(int)
df['episodeNumber'] = df['episodeNumber'].apply(lambda x: 1 if x < 1 else x)
df['episodeNumber'] = np.log10(df['episodeNumber'])
```



```
df[num_cols] = df[num_cols].fillna(df[num_cols].median())
```

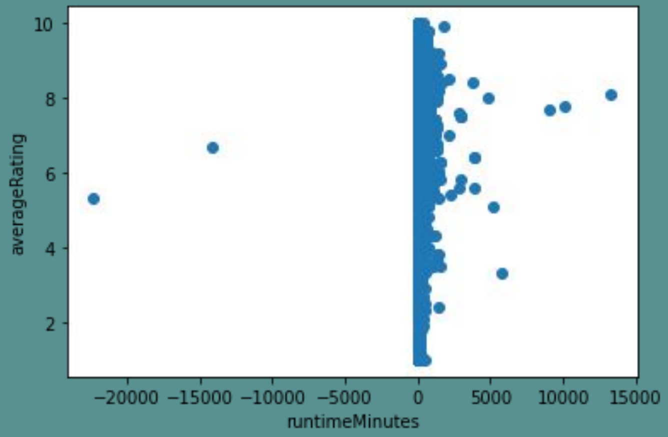


```
df['titleType'] = df['titleType'].replace('tvEpisode', 'tv')
df['titleType'] = df['titleType'].replace('tvMiniSeries', 'tv')
df['titleType'] = df['titleType'].replace('tvSeries', 'tv')
df['titleType'] = df['titleType'].replace('tvSpecial', 'tv')


df['titleType'] = df['titleType'].replace('tvShort', 'short')
df['titleType'] = df['titleType'].replace('tvMovie', 'movie')
```

Oportunidades de mejora:

- Seguir agregando/optimizando features como:
  - Separar por serie completa vs capítulos
  - Encontrar mejores valores para isTop
- Mejores imputaciones que la mediana
- Mejor optimización de hiperparámetros


averageRating
runtimeMinutes