

SEGUNDO EXAMEN PARCIAL

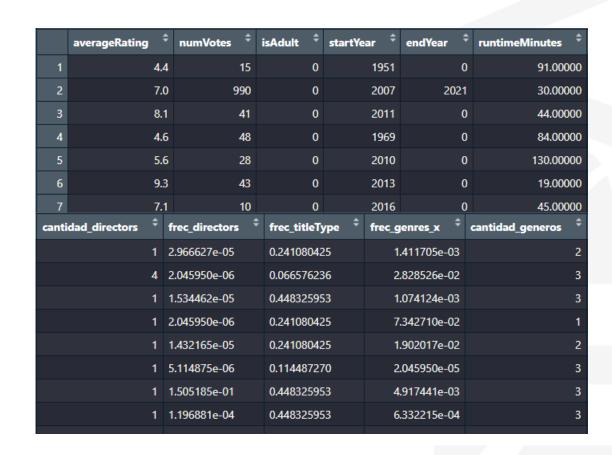
COMPETENCIA DE KAGGLE

_



Limpieza de los datasets

- Fue realizada a través de Rstudio.
- **19 columnas** en el dataset **origen** y **18 columnas** en el dataset de **testeo**.
- Para las variables categóricas realicé **frequency encoding** para pasarlos a datos numéricos.
- Los NAs fueron imputados con las medias de las variables numéricas.
- Los datos de runtimeMinutes en cero fueron reemplazados por la media de la variable.





Limpieza de los datasets

Quedaron los siguientes **features**:

Originales	Nuevos
numVotes	cantidad_writers
isAdult	cantidad_directors
startYear	frec_directors
endYear	frec_titleType
runtimeMinutes	frec_genres_x
seasonNumber	cantidad_géneros
episodeNumber	frec_ori_language
popularity	frec_companies
	duration



Modelo Baseline

- Elegí el modelo de regresión lineal para usar como baseline.
- Tuvo un score de 0.08

```
import os
  import pandas as pd
  import numpy as np
  from sklearn import linear_model
  ✓ 9.1s
```

```
#Creo un modelo regresion lineal
modelo_regresion = linear_model.LinearRegression()
# Entrenar el modelo
modelo_regresion.fit(X,y)
# Realiza las predicciones del conjunto de entrenamiento
Yest = modelo_regresion.predict(X)
```

```
#Baco el score
modelo_regresion.score(X,y)

[8] ✓ 0.2s

... 0.08411775230387053
```



Prueba de XGBOOST

- 1. Empecé probando un modelo XGBOOST sin hiperparámetros.
- 2. Me dio un score de -0.011 que estaba asociado al frecuency encoding que realicé sobre las variables categóricas.
- 3. Una vez corregido eso, obtuve un score de 0.33.
- 4. Fui agregando hiperparámetros y probando distintos valores en ellos. De todas las pruebas, el mayor score que me lanzó fue un 0.38.



Modelo final con RandomForest

- 1. Empecé probando un modelo RandomForest sin hiperparámetros.
- Me dio un score de 0.42
- 3. Fui agregando hiperparámetros y probando distintos valores en ellos. De todas las pruebas, el mayor score que me lanzó fue un 0.43.

```
hiperparametros = {
    'n_estimators': 100,  # Número de árboles en el bosque
    'max depth': None,
    # Profundidad máxima de los árboles (None implica que los árboles se expandirán hasta que todas las hojas contengan menos de min_samples_split muestras)
    'min_samples_split': 3,  # Número mínimo de muestras requeridas para dividir un nodo interno
    'min_samples_leaf': 1,  # Número mínimo de muestras requeridas para estar en un nodo hoja
    'max_features': 'sqrt',  # Cambiado de 'auto' a 'sqrt' (puedes usar también 'auto' o algún número entero)
    'random_state': 42  # semilla para reproducibilidad
}

regresion = RandomForestRegressor(**hiperparametros)

regresion.fit(X_train, y_train)
score = regresion.score(X_test, y_test)

print("R^2 score on testing data: {:.4f}".format(score))

✓ 2m 47.1s

Pyth

R^2 score on testing data: 0.4325
```



Limitaciones y posibles mejoras

- Tardé mucho tiempo en limpiar la base intentando de trabajar con los datos en formato json.
- La próxima vez intentaría limpiar la base de datos directamente en el entorno de Python, para ahorrar tiempo en el ida y vuelta entre los programas y el tiempo de la subida de datos en VisualStudio Code.
- Dediqué dos días completos a la configuración de un código para la búsqueda de hiperparámetros óptimos que se adapten a mi
 modelo. Sin embargo, el proceso no se completó satisfactoriamente y quedó inconcluso.
- En caso de disponer de tiempo adicional, habría profundizado en la investigación sobre la aplicación de un modelo Support Vector Machine.



¡MUCHAS GRACIAS!

MÁS INFORMACIÓN > www.itba.edu.ar