

# Proyecto Final

## MDS7202 - Consultoría a Don Rene

Nosotros:

- Cristóbal Alcázar
- Gianina Salomó



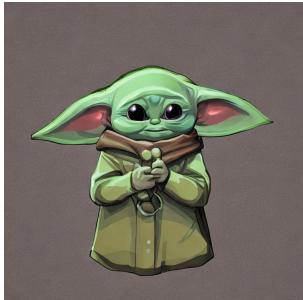
# ¿Qué veremos hoy?

01 EDA

02 Procesamiento

03 Modelos y resultados

04 Conclusiones

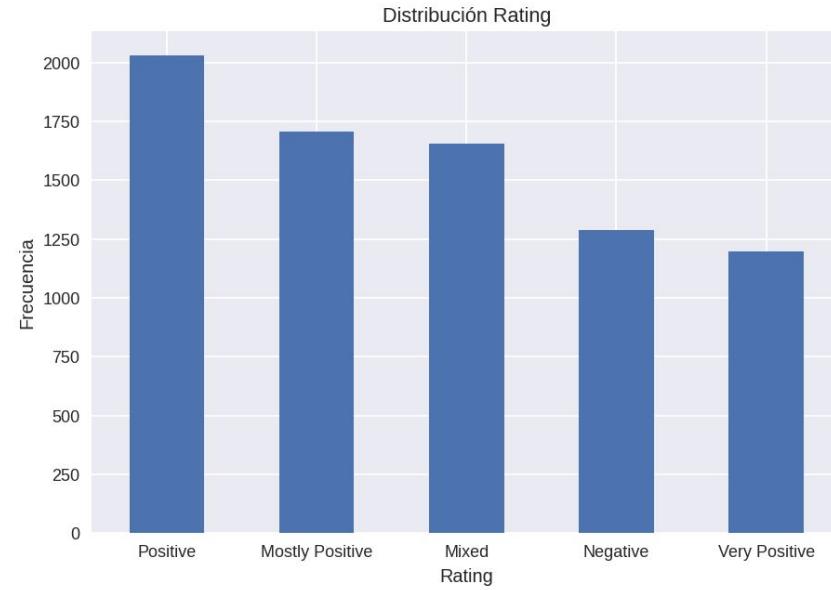
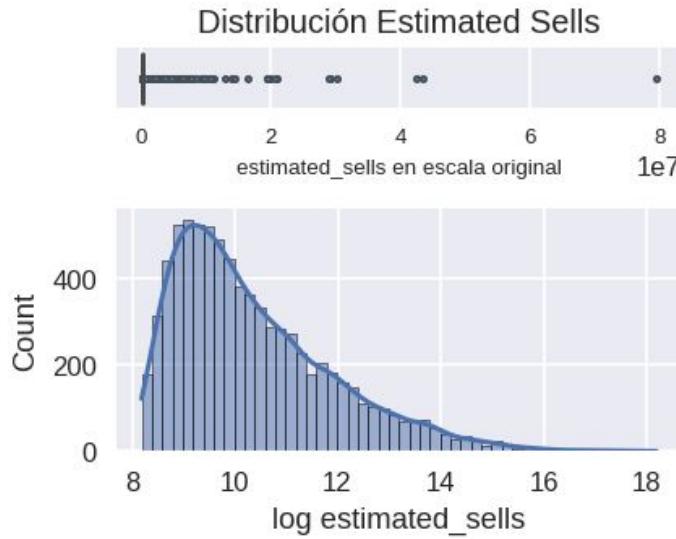




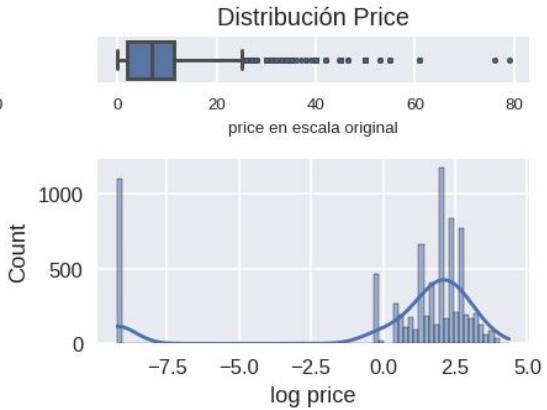
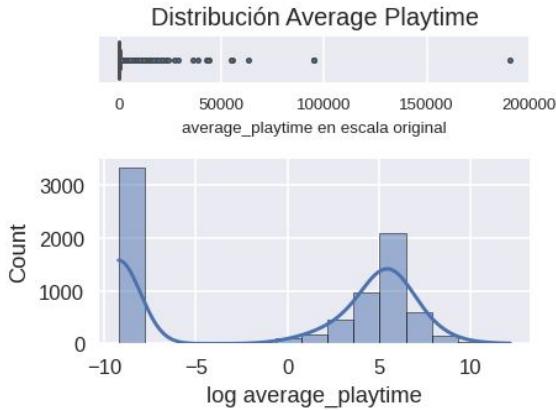
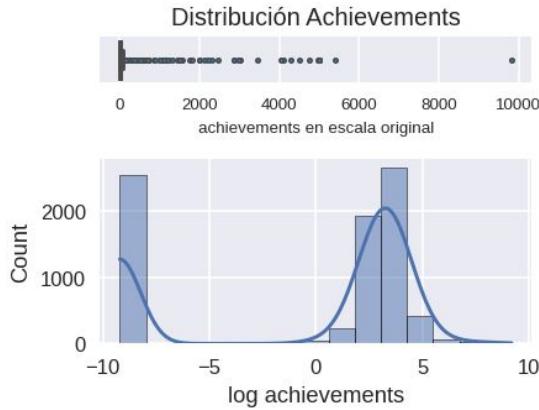
# 1.0

## Análisis Exploratorio de Datos

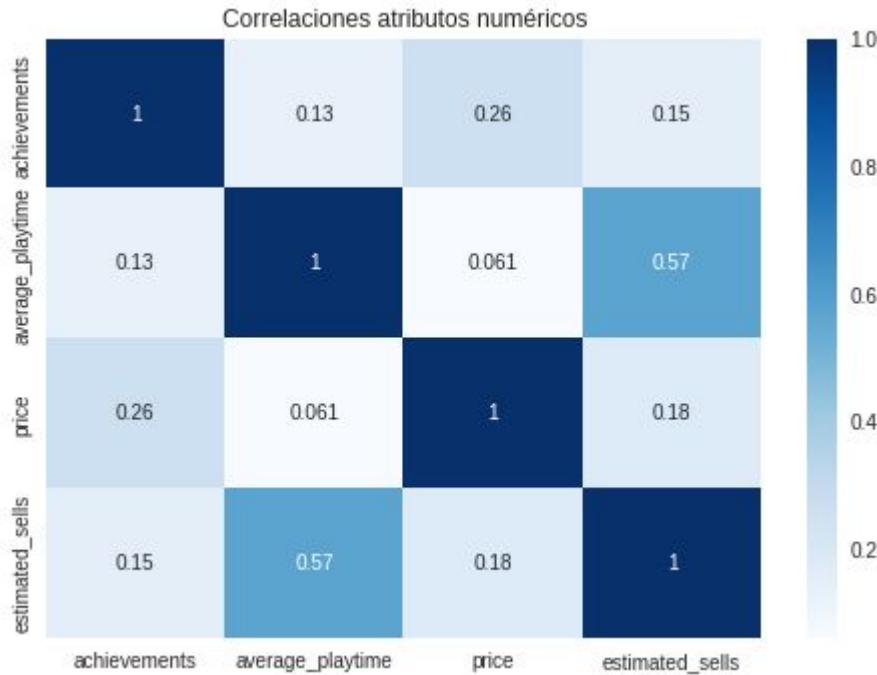
**Ventas tiene una distribución sesgada, y rating es relativamente balanceado**



# Los atributos numéricicos tienen gran cantidad de outliers y distribuciones sesgadas

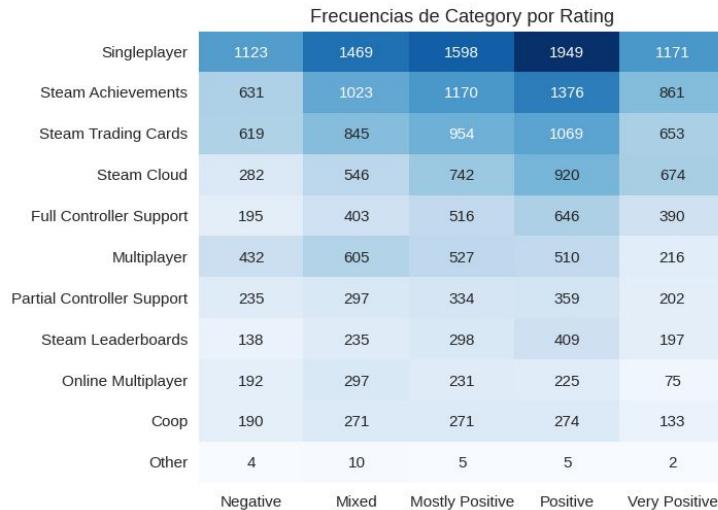
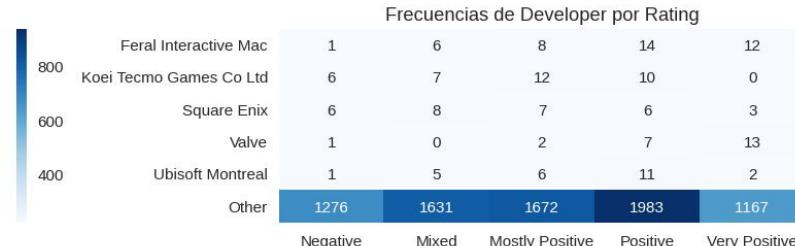
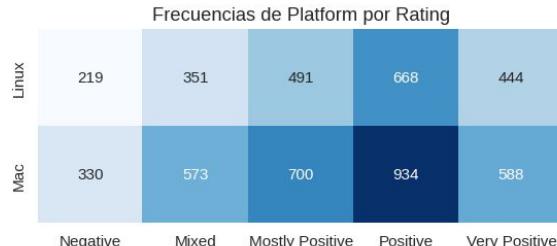


# Baja correlación entre atributos y con ventas



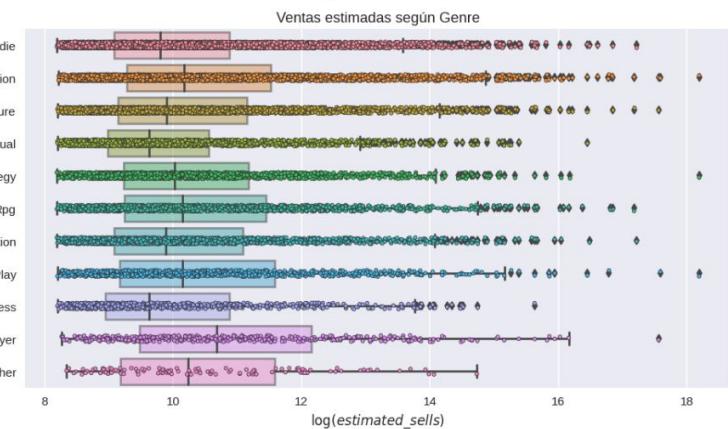
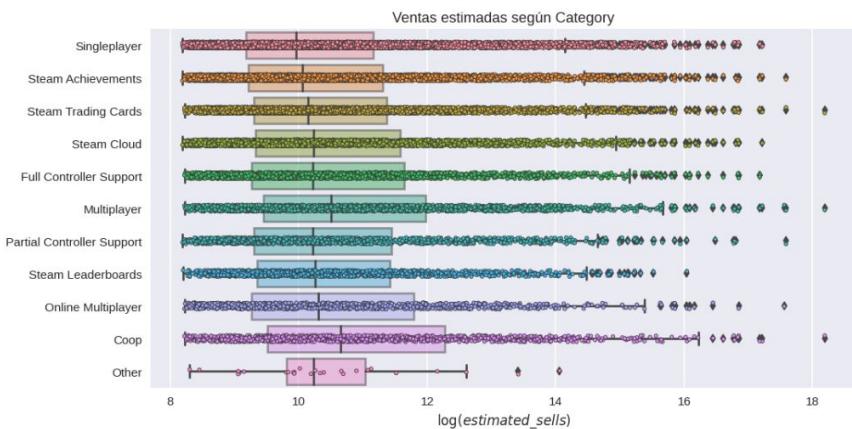
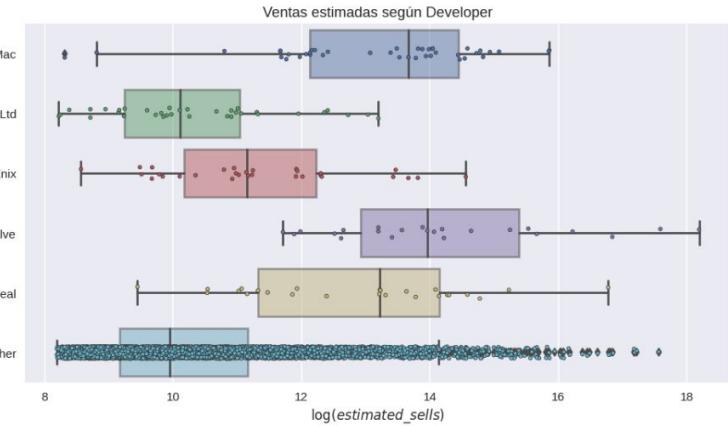
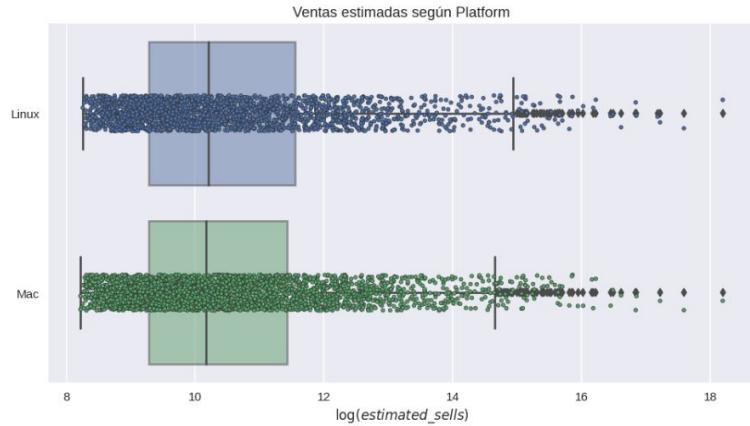
# Developer no es relevante para rating

Tienen mayor relevancia las categorías y géneros más frecuentes

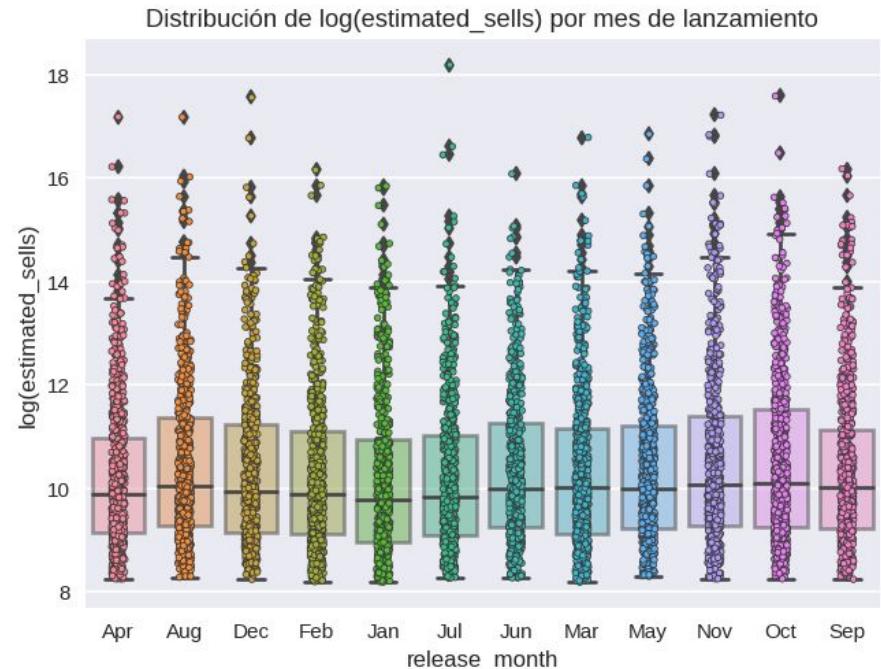
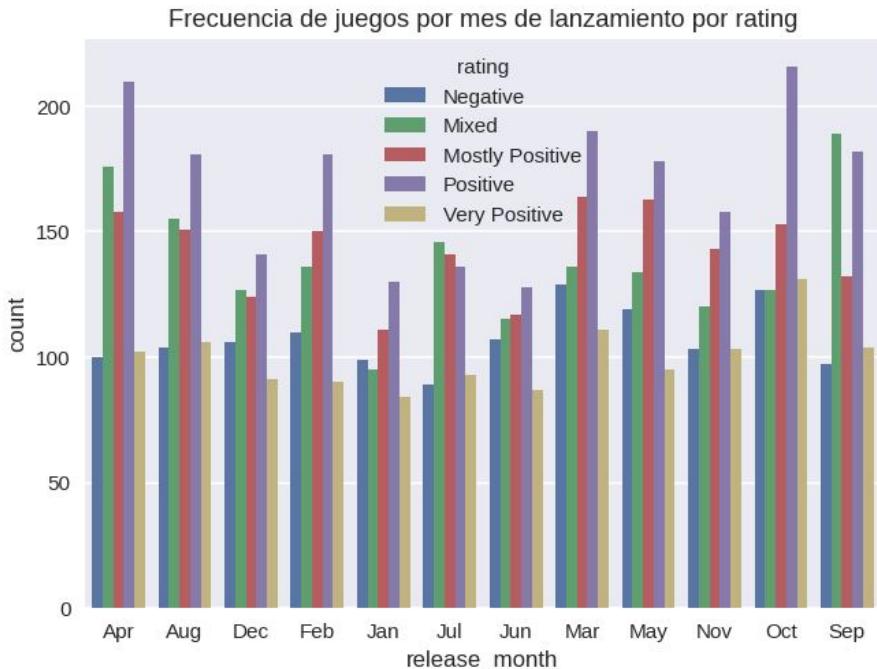


# Plataforma no es relevante para ventas

Diferencias de ventas por developer pero con pocos casos



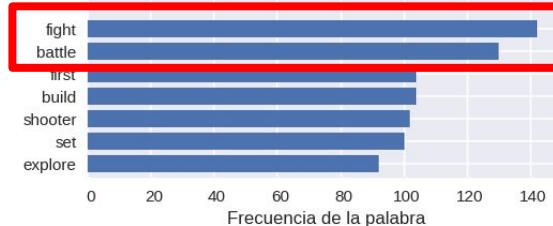
# El mes es relevante para rating pero no para ventas



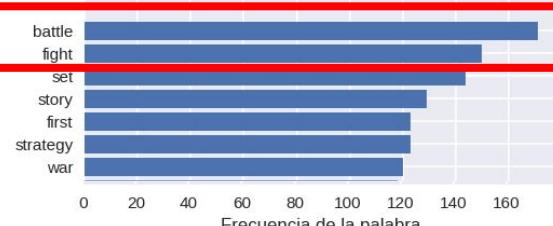
# Mejor calificación juegos descritos con puzzle/story

Peores calificaciones en juegos descritos como de pelea / batalla

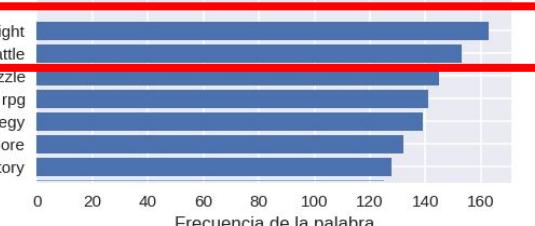
Negative



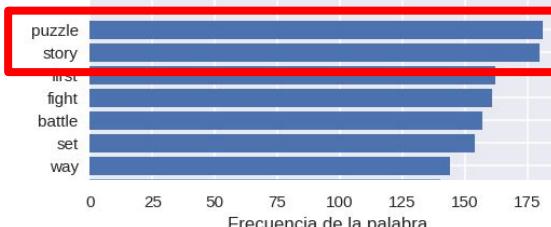
Mixed



Mostly Positive



Positive



Very Positive



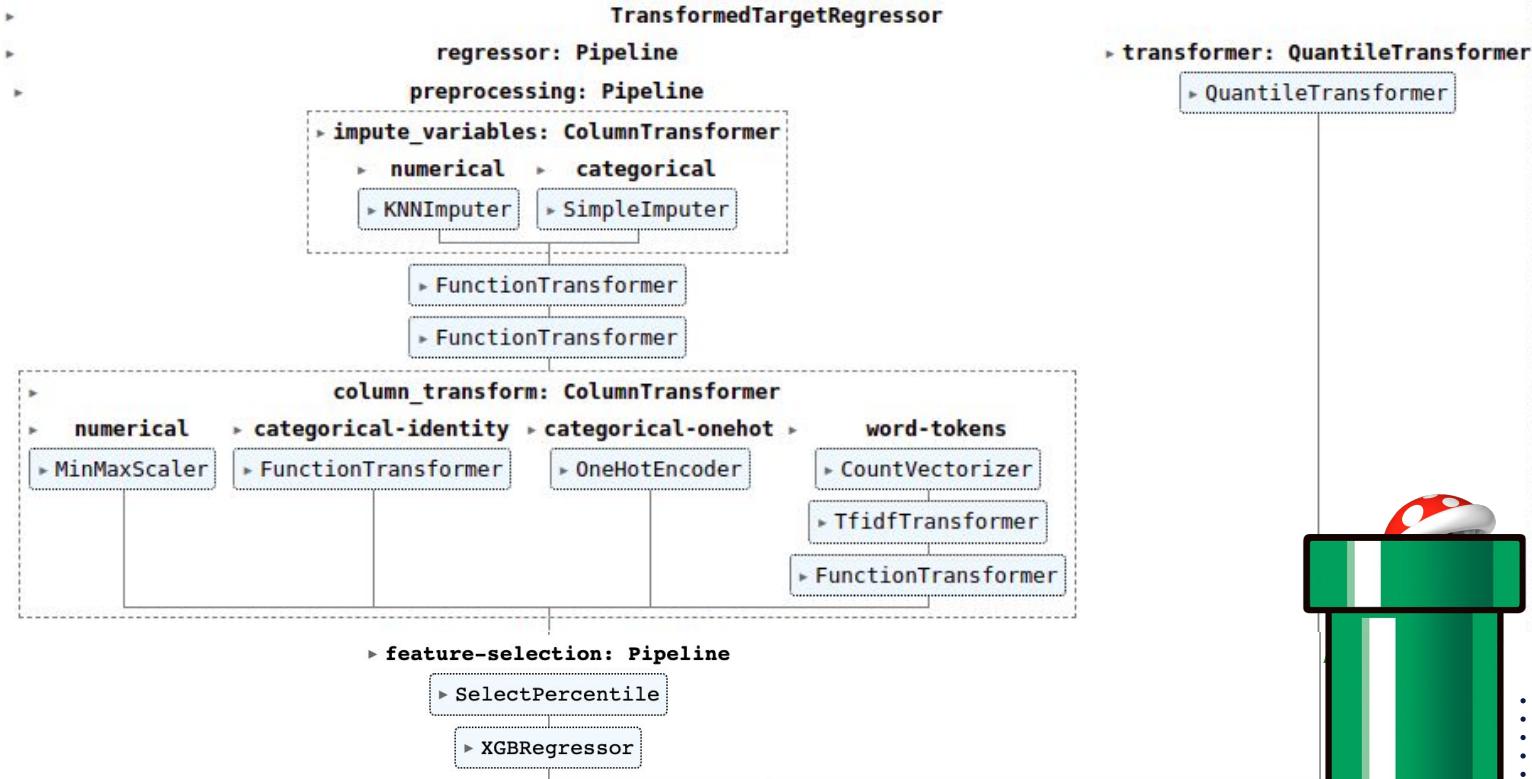
# 2.0

## Procesamiento

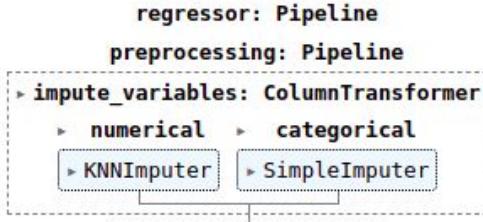
Construcción del *pipeline*



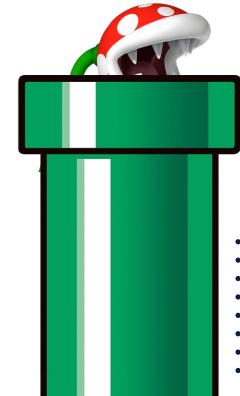
# Nuestro *pipeline* completo:



# 1. Imputar variables: lo que no está, debe estar (robustness, versículo 2) 🤔

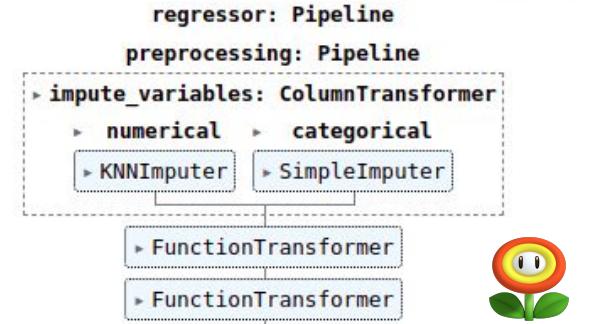


- Observaciones con valores nulos en variables numéricas son imputados por el promedio de sus 3 vecinos más cercanos (**KNNImputer**)
- Observaciones con categorías nulas se les asigna la más frecuente (**SimpleImputer**)

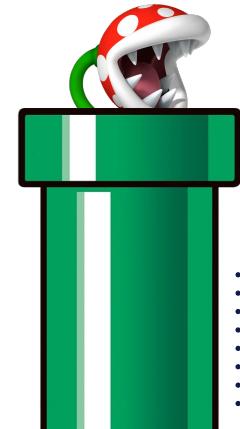


## 2. Funciones de preprocessamiento:

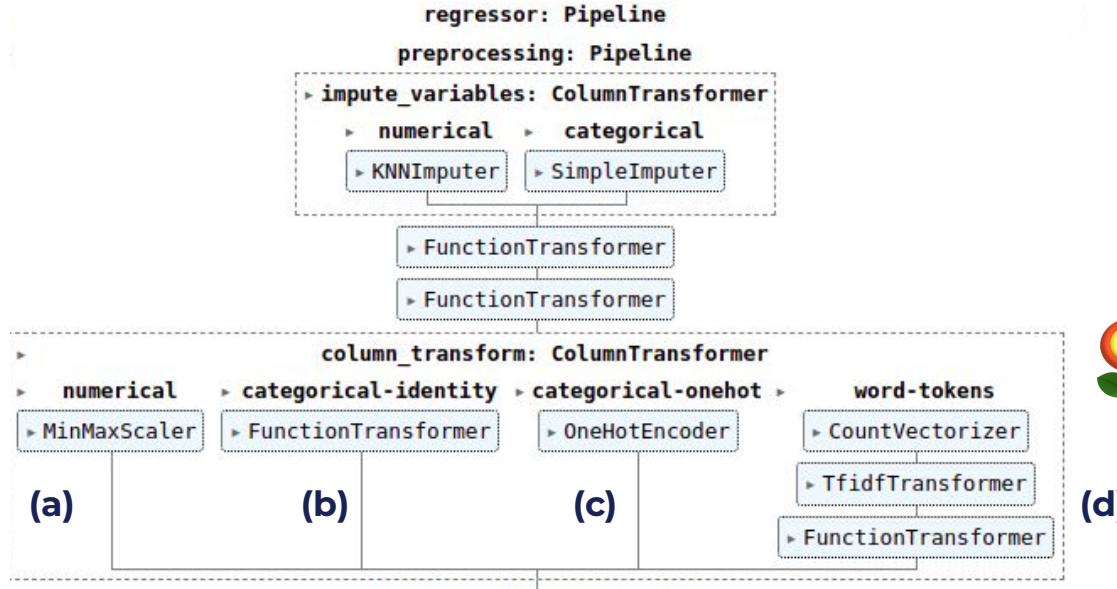
garantizar un mínimo de estructura



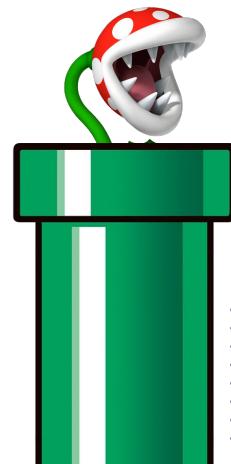
- Columnas categóricas con múltiples etiquetas (e.g. “platform windows;mac;linux”)
- Variables con muchas categorías con baja cardinalidad (i.e. sparse). Selección de las más frecuentes



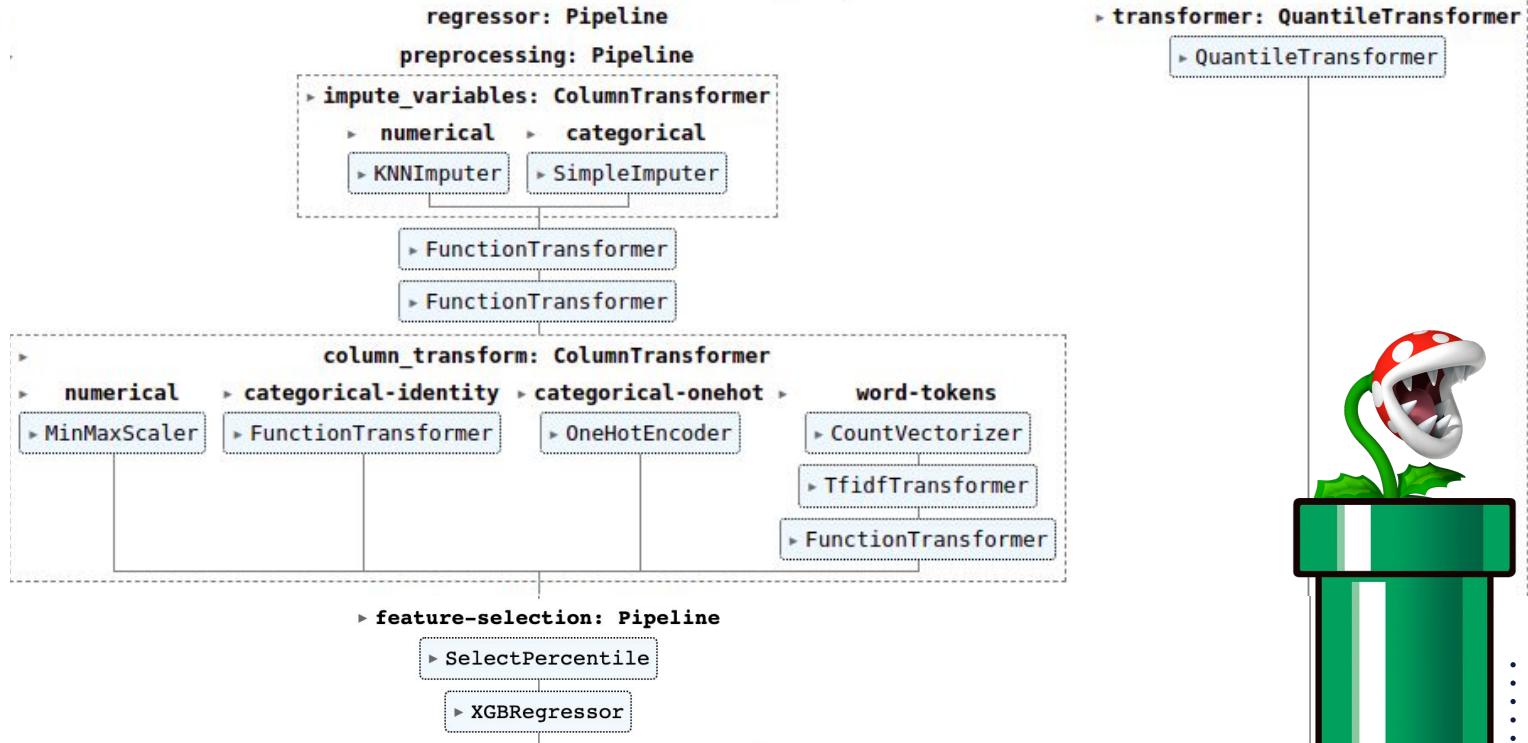
### 3. Transformación de columnas: escalar (a), identidad (b), one-hot encoding (c), y bag-of-words (d)



(d)



# 4. Transformación sobre los targets: en caso problema de regresión se incorpora QuantileTransformer

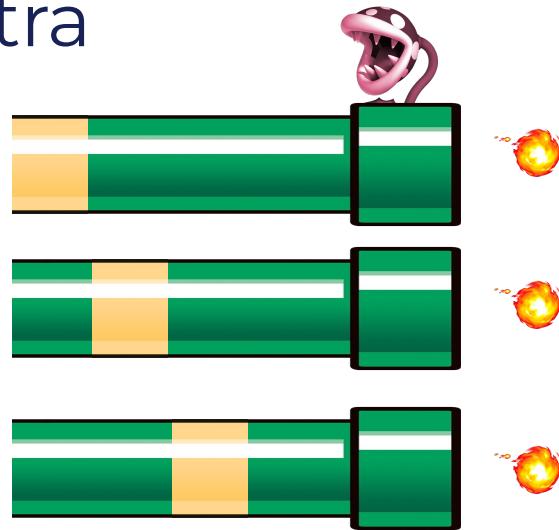


# 5. Separación de conjuntos para entrenar y evaluar fuera de la muestra

Evaluación (1.5% - 119 observaciones) (\*)



Entrenamiento  
(98.5% - 7.762 observaciones)



Entrenamiento  
con K-Fold Cross  
Validation ( $cv=5$ )



(\*) Nuestra estrategia fue evitar sustraer la mayor cantidad de observaciones posibles del conjunto de entrenamiento dado que usamos K-Fold Cross Validation. No eran muchos datos.



# 3.0

## Modelos y resultados

Train, Test y Validación.



# Usamos de baseline



*Los 3 vecinos más cercanos para votar por mayoría (clasificación) o promediar (regresión)*

Modelo	F1 Weighted	R2
DummyClassifier	0.2345	-
KNeighborsClassifier(n_neighbors=3)	0.2359	-
DummyRegressor	-	-0.08
KNeighborsRegressor(n_neighbors=3)	-	0.15

# Estrategia de entrenamiento 💪

01 Optuna Amplio

02 GridSearchCV acotado

03 Evaluación mejor GridSearch



# Clasificación de rating ⭐

*F1-Weighted*



Modelo	Optuna	GridSearch*	Train**	Test**	Competencia**
RFC	0.30	0.31	-	-	-
XGBC	0.33	0.31	-	-	-
SVC	0.34	0.33	0.45	0.34	0.30

(\*) Mejor F1-Weighted en validación

(\*\*) Utilizando el mejor modelo de GridSearch

# Regresión de ventas



$R^2$

Modelo	Optuna	GridSearch*	Train**	Test**	Competencia**
ElasticNet	- 0.018	-	-	-	-
XGBR	<b>0.2578</b>	<b>0.26</b>	<b>0.19</b>	<b>0.16</b>	<b>0.28</b>
HubberBagging	0.0368	-	-	-	-

(\*) Mejor  $R^2$  en validación

(\*\*) Utilizando el mejor modelo de GridSearch

# Modelos mejorados

Optuna

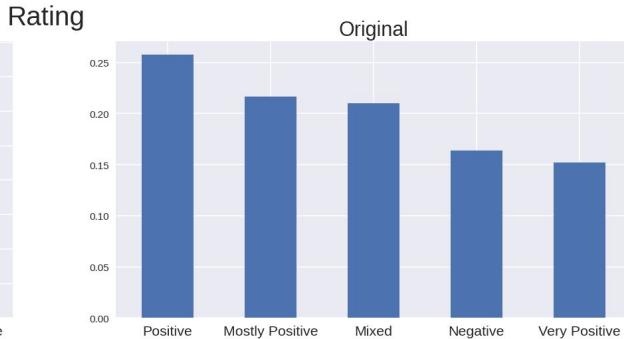
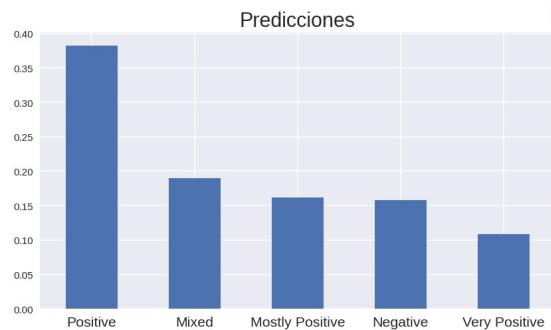
KFold

*Optuna acotado en  
mejor(es) modelo  
con 10 KFold.*

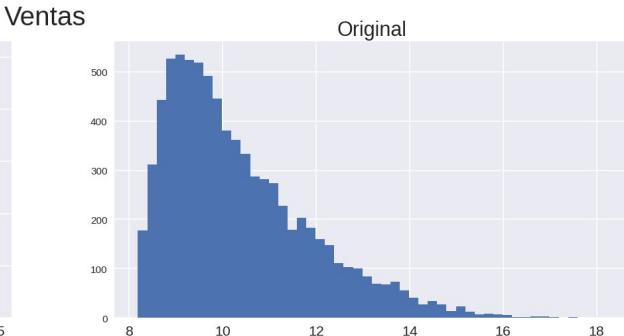
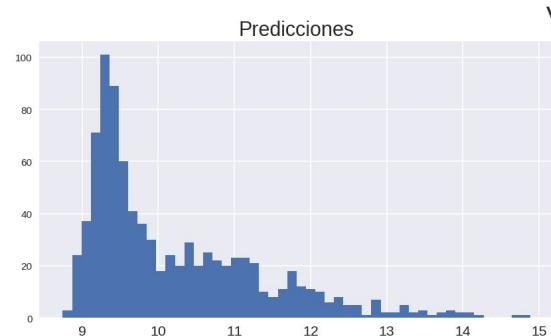
*Mejor modelo Optuna,  
reentrenando y  
probando en 10 KFold.*

*Escoger modelo con  
scores más  
consistentes / altos, y  
enviar a competencia.*

# Mejores intentos



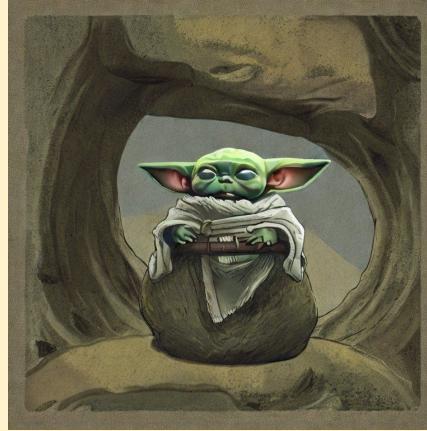
**XGBClassifier**  
F1 weighted 0.32



**XGBRegressor**  
 $R^2$  0.28



# 4.0



## Conclusiones



# Conclusiones



- **Resolución del problema:** Creamos un proceso robusto para pre-procesar, transformar, y experimentar (pipeline + optuna).
- **Baseline:** Uso de modelo simple y fácil de evaluar que permite ir midiendo la efectividad de innovaciones en el proceso de modelamiento. KNN cumple este propósito, siendo un modelo intuitivo en su funcionamiento y simple de correr.
- **Resultados:** Superamos el baseline en  $\geq 10$  pts respecto a r2, y 8 pts en f1-score. Métricas aceptables, pero no conformes. No obstante, el cómo construimos la solución permite fácil buscar mejores resultados. (23.59 fs y 0.15 r2)
- **Aprendizajes:** Optimización de hiper-parámetros con optuna, distintos sampling para explorar el espacio de configuraciones, subir de nivel en uso de sklearn pipelines (e.g. TargetTransform), transformaciones continuas (box-cox, quantile), y wrapper de bagging.
- **Competencia:** Nos pareció buena idea; permite contar con otra medición aparte del baseline, una especie de benchmark de lo que es posible lograr.

# Anexos



# A mayor tiempo de juego, leve alza en ventas.

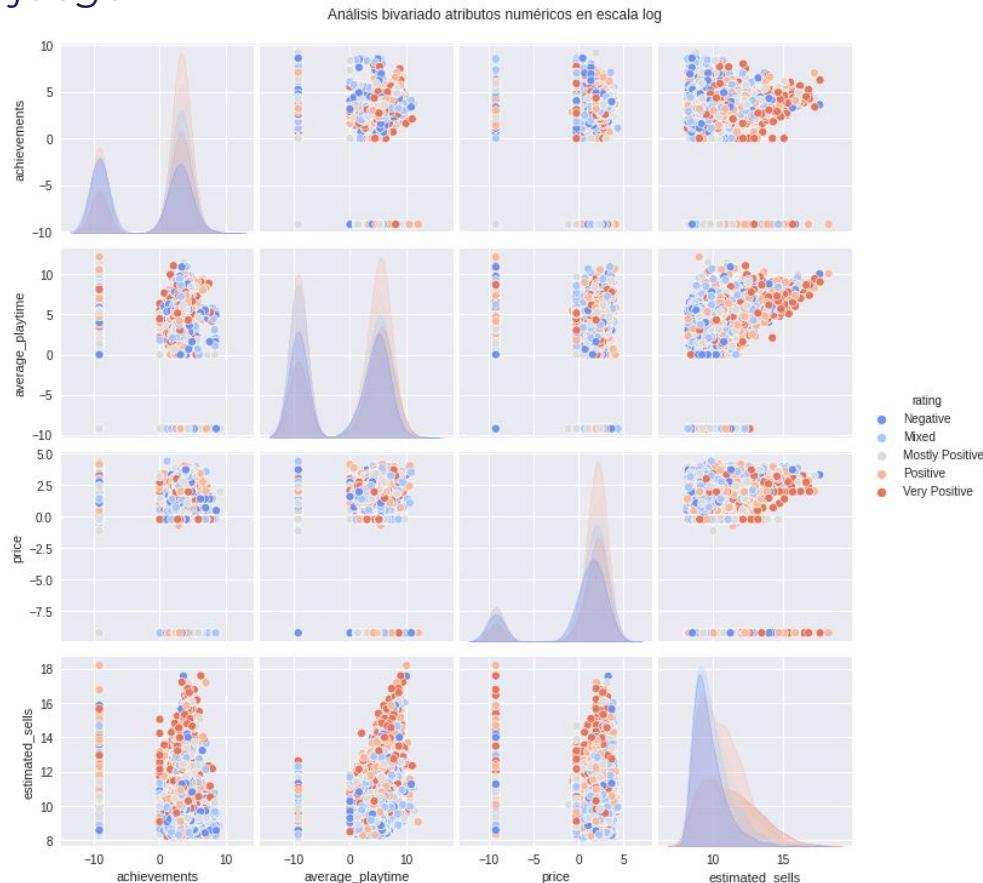
Los juegos con calificación positiva tienen una mayor cantidad de logros, precio, y tiempo de juego

Logros ->

Tiempo promedio de juego ->

Precio ->

Estimado de ventas ->



## Comportamiento variable precio, por categoría tag y rating.

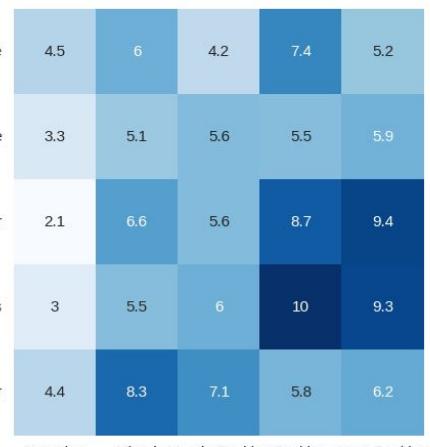
Cada Heatmap corresponde a un subconjunto de datos de género, diferencias entre considerar solo género o grupo genero + tag.

Promedio de la variable price dado el género de juego genre\_indie según los grupos tags y rating



Categorías de rating  
(promedio por categorías de rating solo considerando el género)

Promedio de la variable price dado el género de juego genre\_indie según los grupos tags y rating



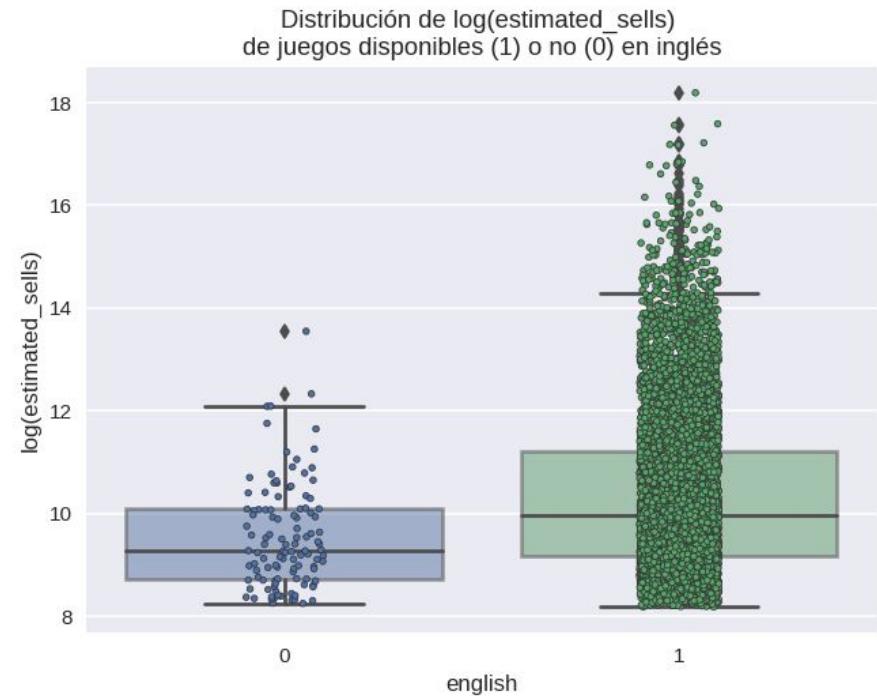
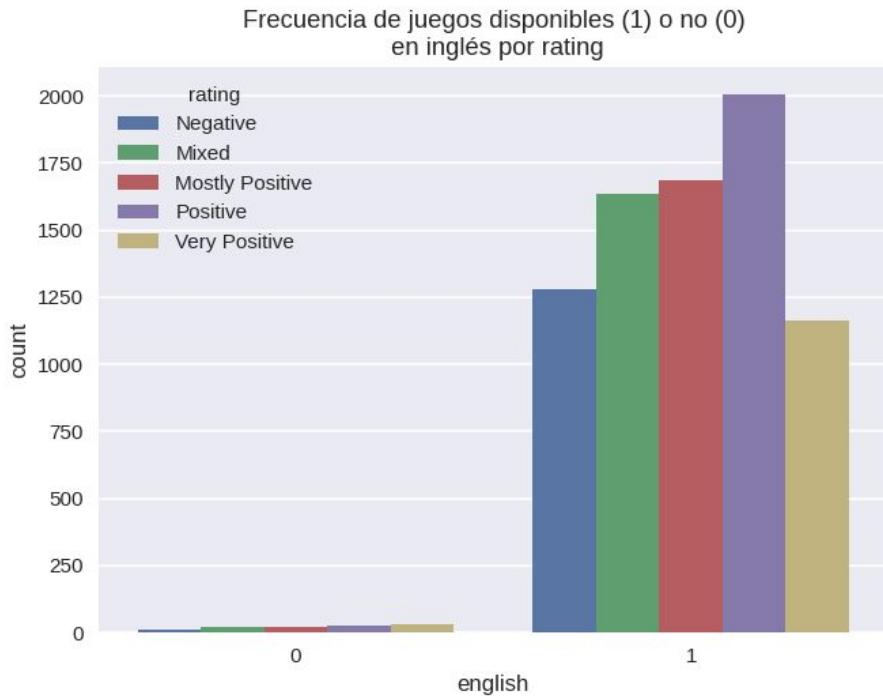
Categorías de rating  
(promedio por categorías de rating solo considerando el género)

Promedio de la variable price dado el género de juego genre\_indie según los grupos tags y rating

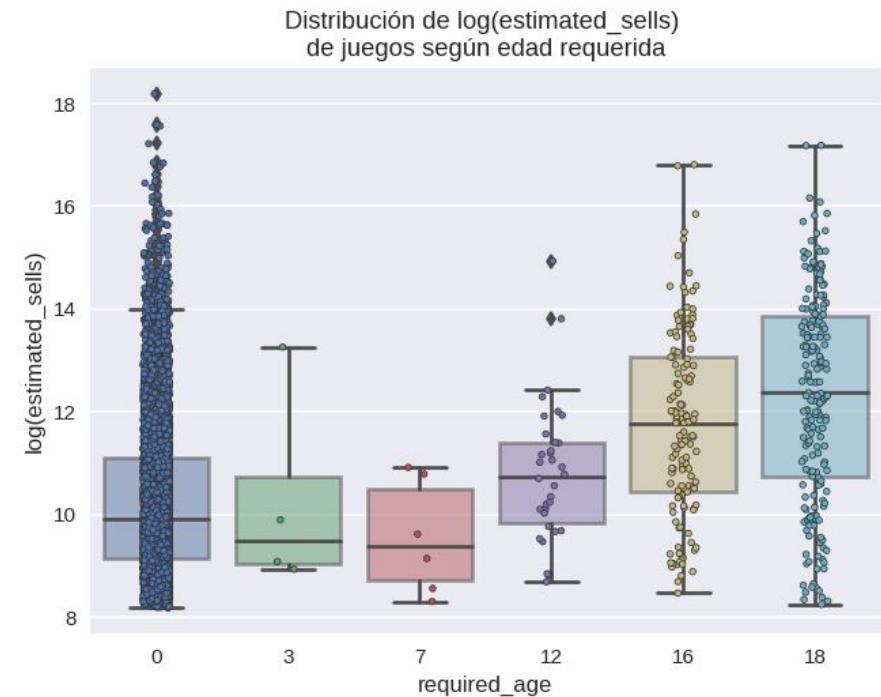
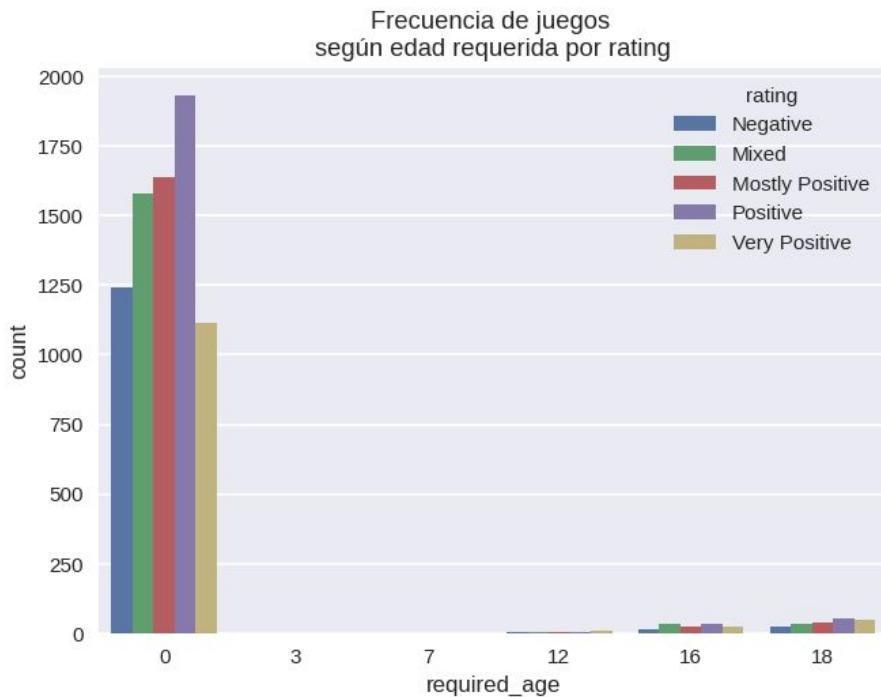


Categorías de rating  
(promedio por categorías de rating solo considerando el género)

# Inglés es relevante para ventas pero no para rating

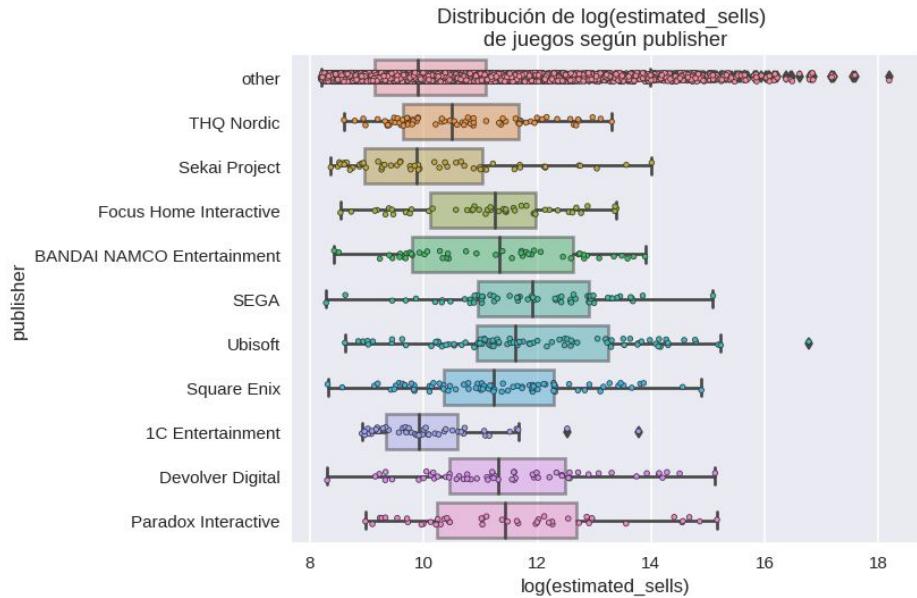
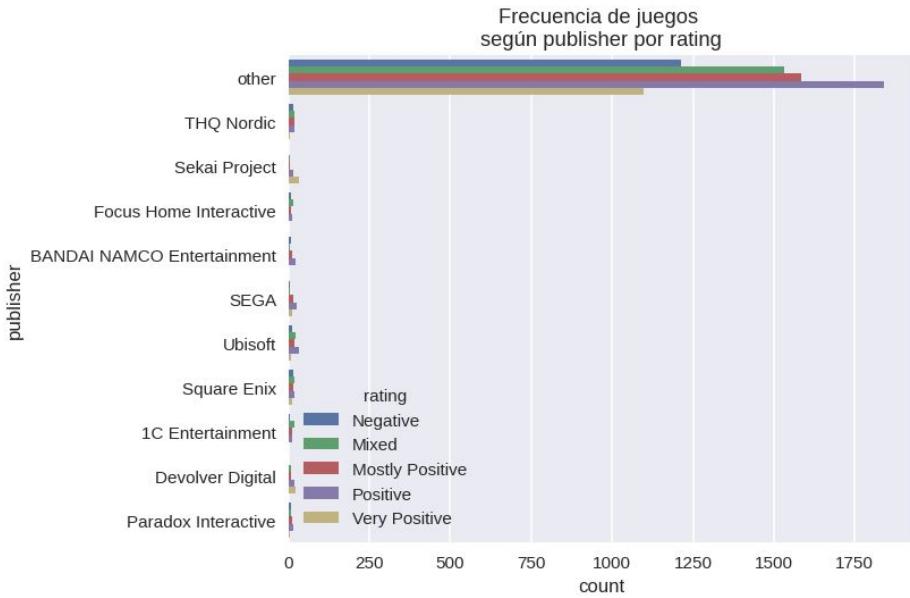


# Edad es relevante para ventas pero no para rating



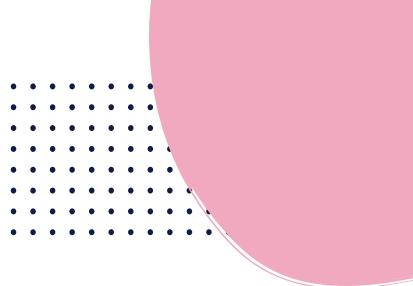
# Muchos publisher diferentes

Algo de relevancia en ventas

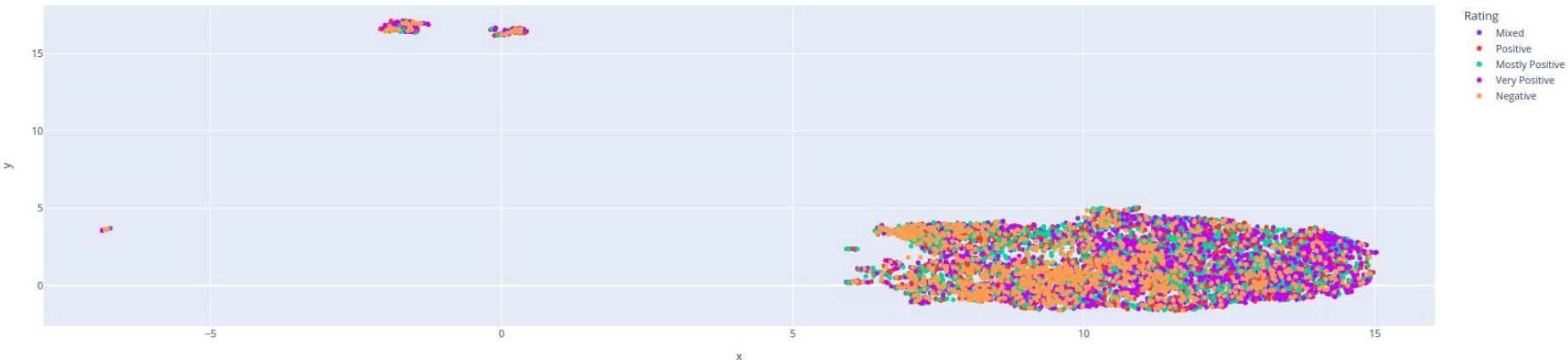


# Valores más altos en proyección x tienen más valoraciones Very Positive

Valores bajos en proyección x tienen una valoración negativa



Proyección UMAP



Rating

- Mixed
- Positive
- Mostly Positive
- Very Positive
- Negative

# Juegos Very Positive tienen valores más altos de las proyecciones x e y

Juegos negativos tienen valores más bajos de estas proyecciones

Proyección TSNE

