El desafío de Don René

Proyecto MDS7202

Johnny Godoy

Agenda

- Introducción al problema
- Análisis exploratorio
- Preparación de datos
- Modelos
- Conclusiones

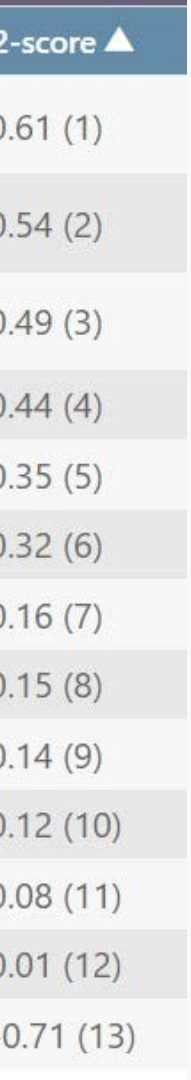
El problema

A predecir

Dado una base de datos cuyos registros son videojuegos, queremos generar modelos que puedan predecir:

- Evaluaciones de jugadores con clasificación
- Ventas potenciales de los juegos con regresión

Se evalúa f1_weighted y r^2



0.61 / #1*

*Se sobreescribió por un puntaje peor por accidente,

bajando a #4

0.38 / #2

Análisis exploratorio

Parte 1: Univariado

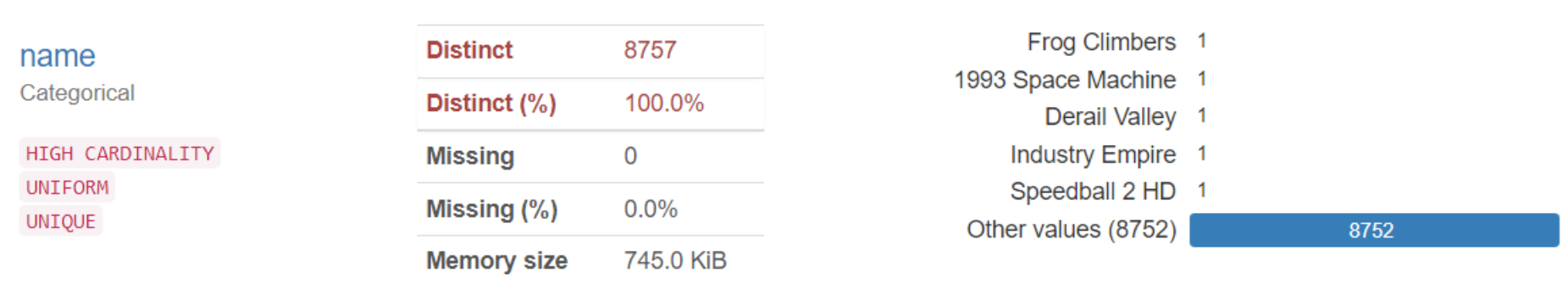
Resumen de los datos

Dataset statistics	
Number of variables	15
Number of observations	8757
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	9.2 MiB
Average record size in memory	1.1 KiB

Variable types	
Categorical	10
Numeric	5

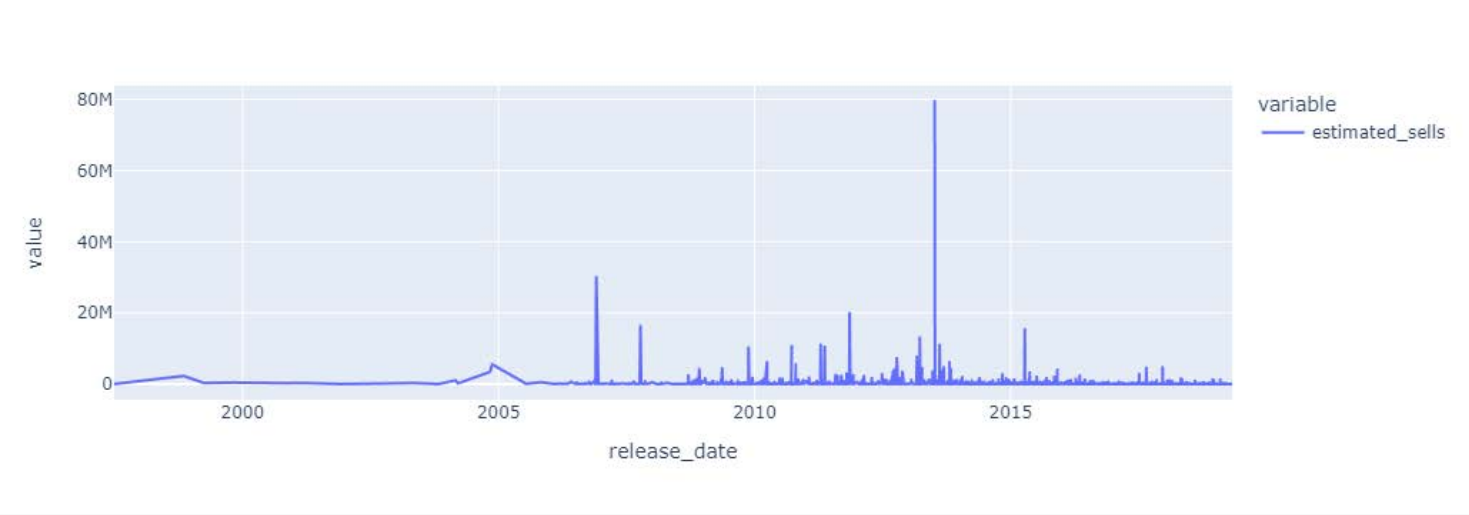
De estos tenemos 7881 de entrenamiento y 876 que predecir, o sea, 10% del total.

Name



Variable identificadora única. No tiene información útil, pues en principio las franquicias famosas aún así representan pocos datos

Release date



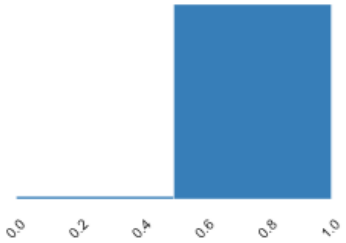
Se divide en 3 características día-mes-año. Se agrega un indicador si estamos cerca de la Navidad, por los descuentos que ocurren en esta fecha.

English

english

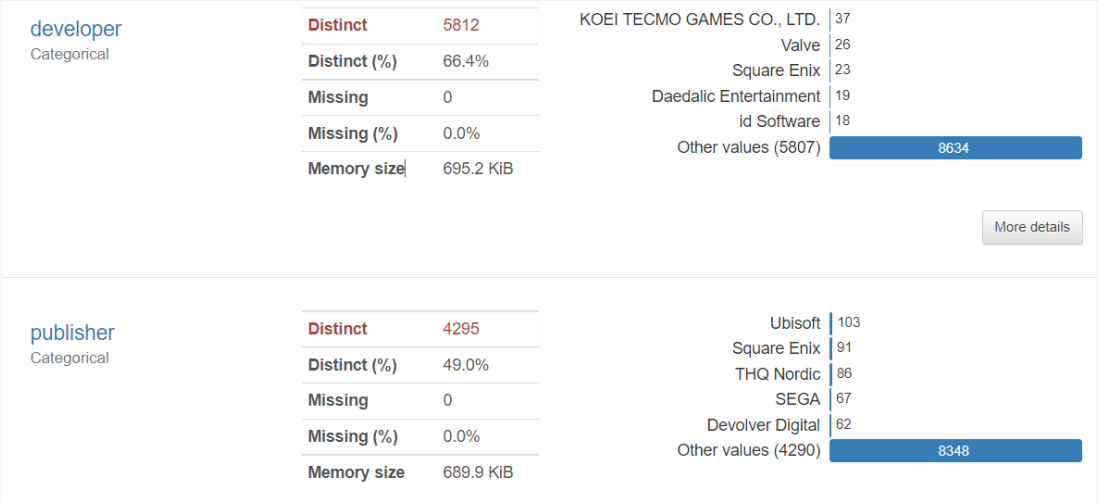
Real number (ℝ)

Distinct	2	**Minimum**	0
Distinct (%)	< 0.1%	**Maximum**	1
Missing	0	**Zeros**	122
Missing (%)	0.0%	**Zeros (%)**	1.4%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	0.98606829	**Memory size**	136.8 KiB



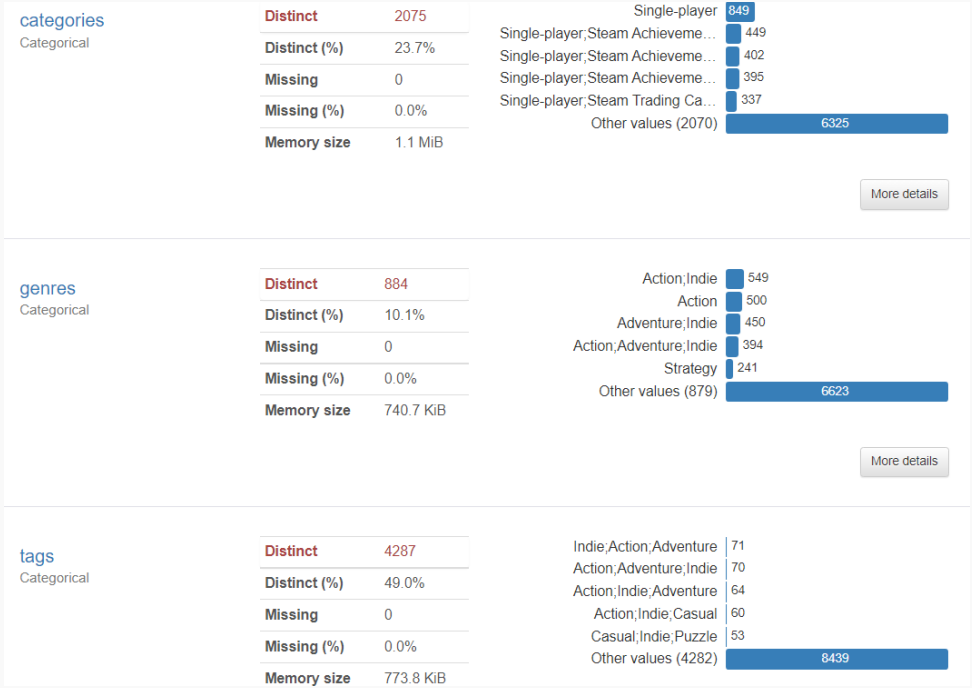
Binaria y desbalanceada, puede tener sentido estratificar.

Developer y publisher



Puede tener sentido crear variables para los pocos famosos que aparecen más, relacionándolos con sus ganancias. Estas variables no se usaron.

Categories, Genres, Tags



categories
Categorical

Distinct	2075
Distinct (%)	23.7%
Missing	0
Missing (%)	0.0%
Memory size	1.1 MiB

Single-player	849
Single-player;Steam Achieveme…	449
Single-player;Steam Achieveme…	402
Single-player;Steam Achieveme…	395
Single-player;Steam Trading Ca…	337
Other values (2070)	6325

More details

genres
Categorical

Distinct	884
Distinct (%)	10.1%
Missing	0
Missing (%)	0.0%
Memory size	740.7 KiB

Action;Indie	549
Action	500
Adventure;Indie	450
Action;Adventure;Indie	394
Strategy	241
Other values (879)	6623

More details

tags
Categorical

Distinct	4287
Distinct (%)	49.0%
Missing	0
Missing (%)	0.0%
Memory size	773.8 KiB

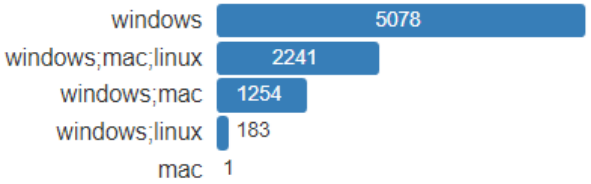
Indie;Action;Adventure	71
Action;Adventure;Indie	70
Action;Indie;Adventure	64
Action;Indie;Casual	60
Casual;Indie;Puzzle	53
Other values (4282)	8439

- Texto separado por ";", se usa como tokenizador para crear atributos con codificación binaria
- A veces estas 3 repiten la misma información, y pueden tener sinónimos, así que reducir dimensionalidad tiene sentido

Platforms

platforms
Categorical

Distinct	5
Distinct (%)	0.1%
Missing	0
Missing (%)	0.0%
Memory size	643.6 KiB

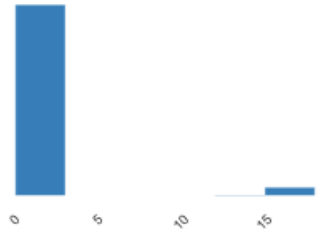


Similar a las anteriores, pero con muchas menos combinaciones y sin problemas como sinonimia, no tiene sentido tratarlo como texto sino que como categorías

Required age

required_age
Real number (ℝ)

Distinct	6	Minimum	0
Distinct (%)	0.1%	Maximum	18
Missing	0	Zeros	8343
Missing (%)	0.0%	Zeros (%)	95.3%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	0.77720681	Memory size	136.8 KiB

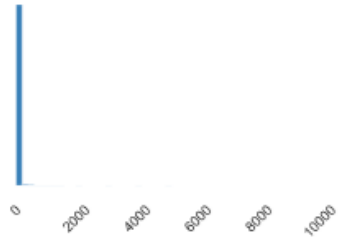


Numérica con 6 valores únicos, así que discreta.

Achievements

achievements
Real number (ℝ)

Distinct	290	**Minimum**	0
Distinct (%)	3.3%	**Maximum**	9821
Missing	0	**Zeros**	2812
Missing (%)	0.0%	**Zeros (%)**	32.1%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	43.226219	**Memory size**	136.8 KiB



Muy concentrada en 0.

Average playtime

average_playtime

Real number (R)

SKEWED

ZEROS

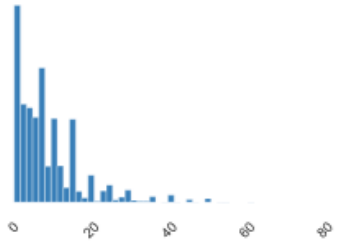
Distinct	1326	**Minimum**	0
Distinct (%)	15.1%	**Maximum**	190625
Missing	0	**Zeros**	3681
Missing (%)	0.0%	**Zeros (%)**	42.0%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	423.30273	**Memory size**	136.8 KiB

Muy concentrada en 0, lo cual es sospechoso pues este valor debería ser inválido, y puede ser un valor imputado en caso de que no se tenga información.

Price

price
Real number (ℝ)

Distinct	168	**Minimum**	0
Distinct (%)	1.9%	**Maximum**	78.99
Missing	0	**Zeros**	1216
Missing (%)	0.0%	**Zeros (%)**	13.9%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	8.4263869	**Memory size**	136.8 KiB



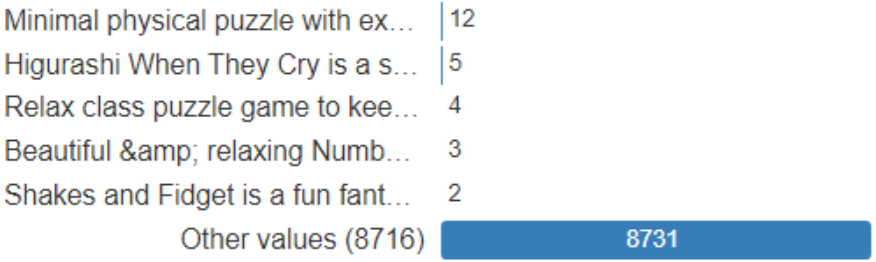
Una cantidad decente de ceros (juegos gratis), pero que baja de manera más esperada después. Una transformación logarítimica puede tener sentido.

Short description

short_description
Categorical

HIGH CARDINALITY
UNIFORM

Distinct	8721
Distinct (%)	99.6%
Missing	0
Missing (%)	0.0%
Memory size	2.7 MiB



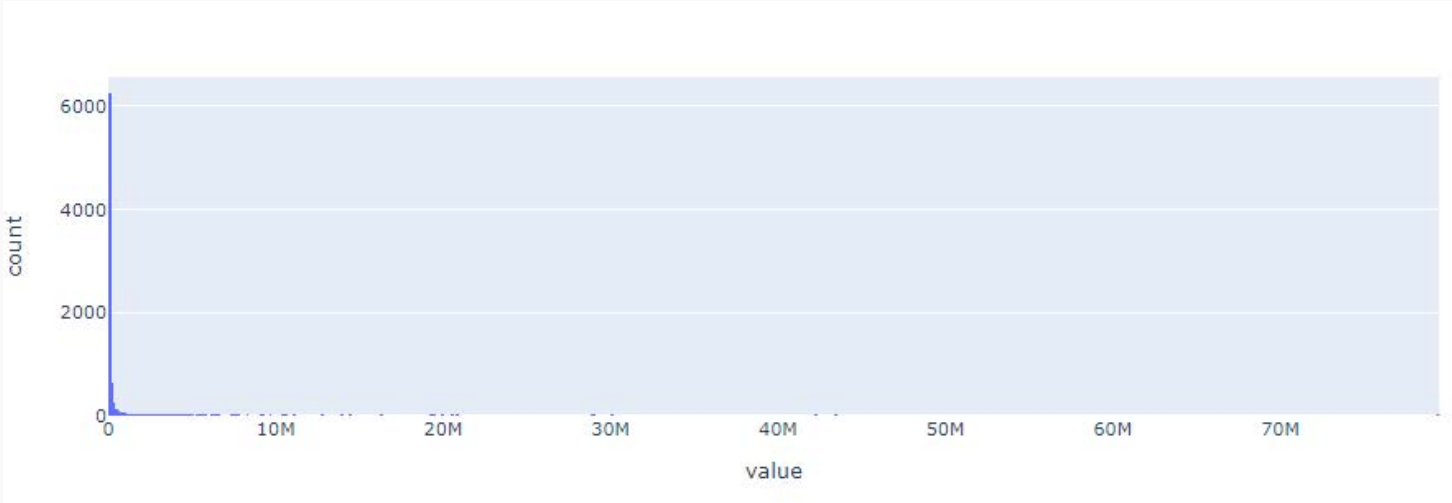
Variable totalmente textual que sí puede ameritar usar técnicas de NLP más avanzadas. Es raro que tenga duplicados.

Claim sin base: La gente no lee las descripciones del juego para decidir sobre su compra

Variables de respuesta: Rating

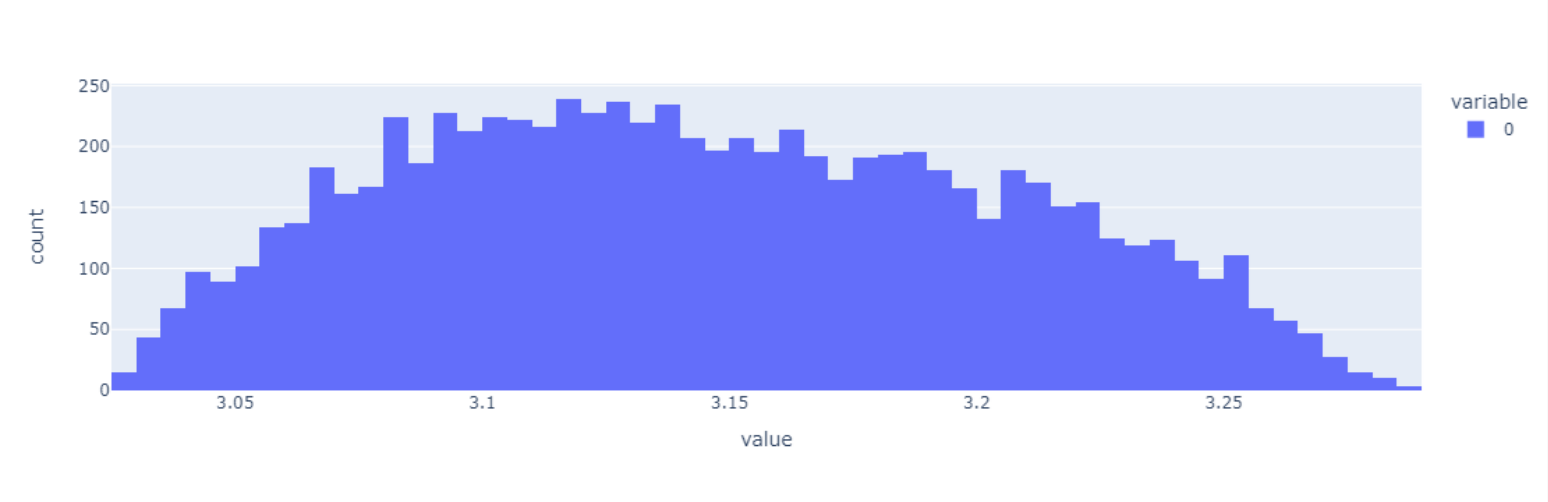
Todas las clases están bien balanceadas. Existe una relación de ordinalidad que se podría explorar.

Variables de respuesta: Ventas estimadas



¡No se vé nada! A aplicar una transformación de Box Cox

Variables de respuesta: Ventas estimadas

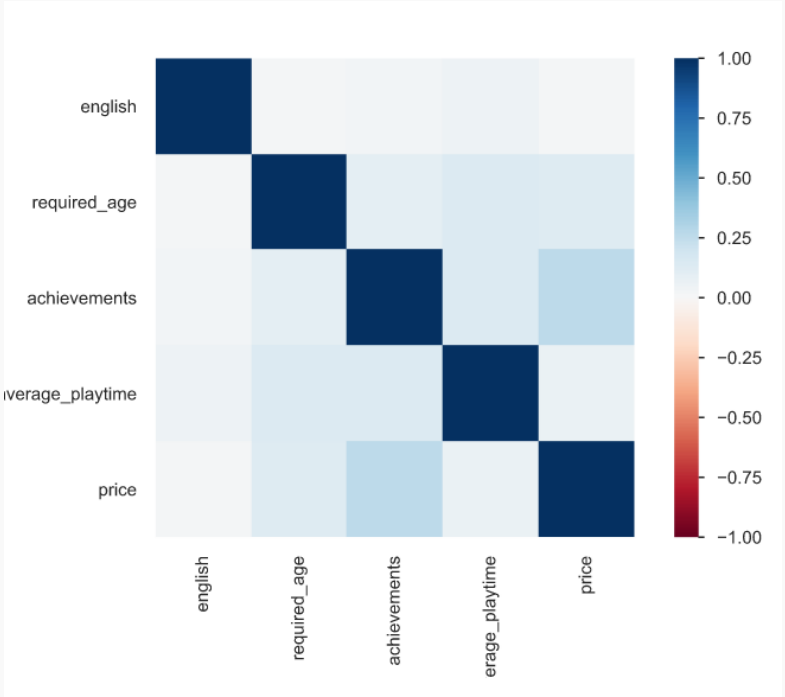


Mucho mejor. Esta transformación ayuda a los siguientes gráficos, pero empeora las predicciones.

Análisis exploratorio

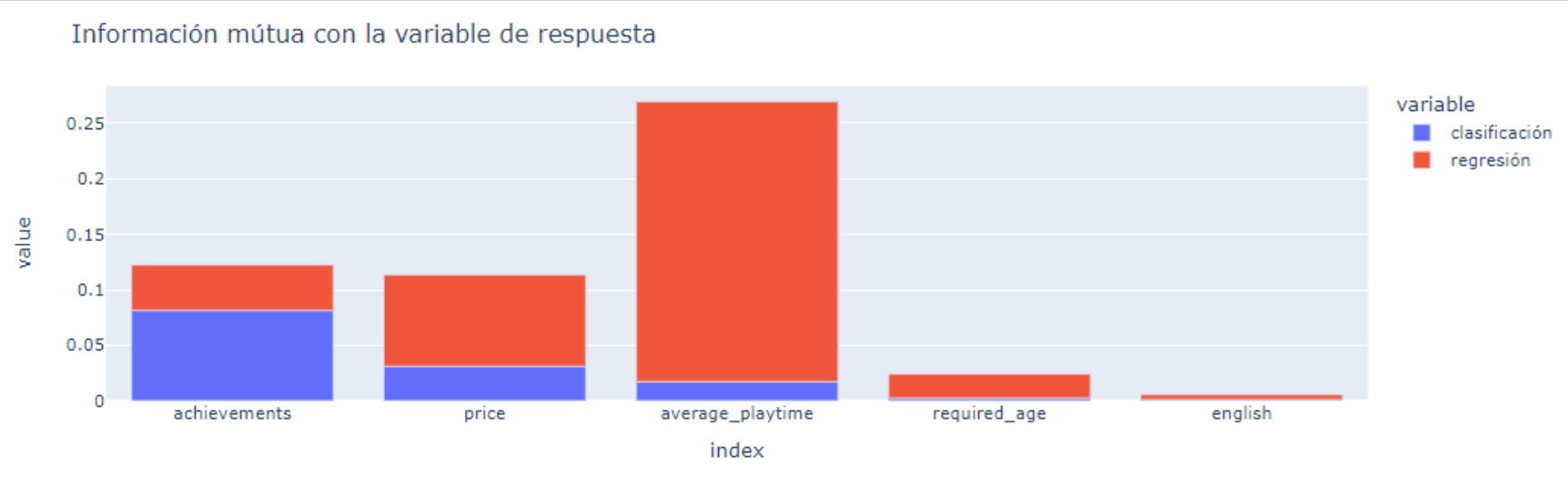
Parte 1: Bivariado

Matriz de correlación



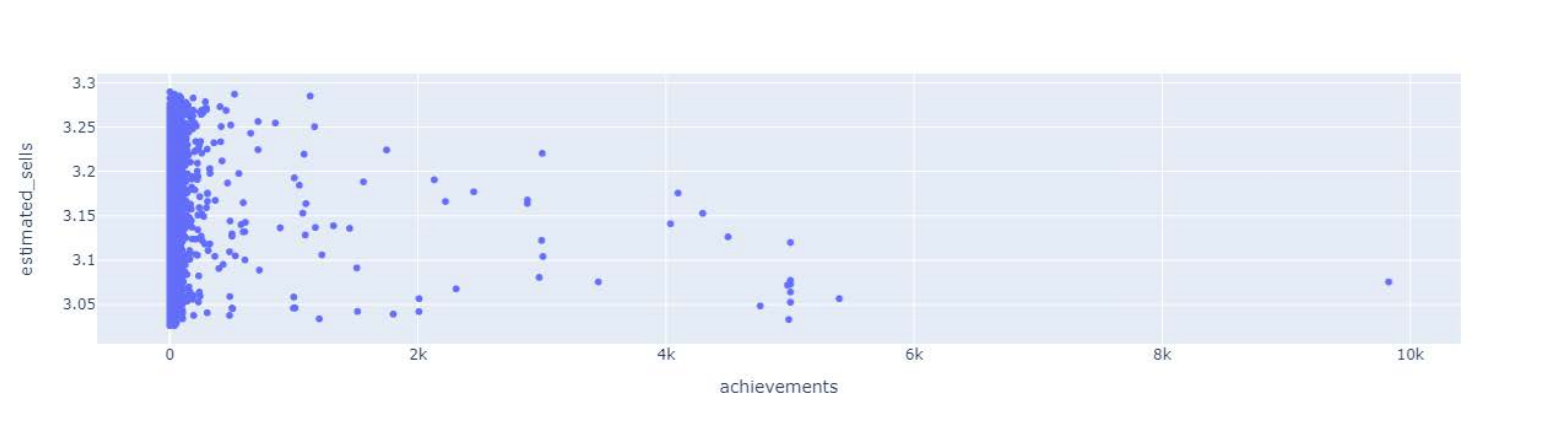
- La única no nula no trivial es achievements vs price
- Juegos más producidos (más logros) cuestan más

Información mútua



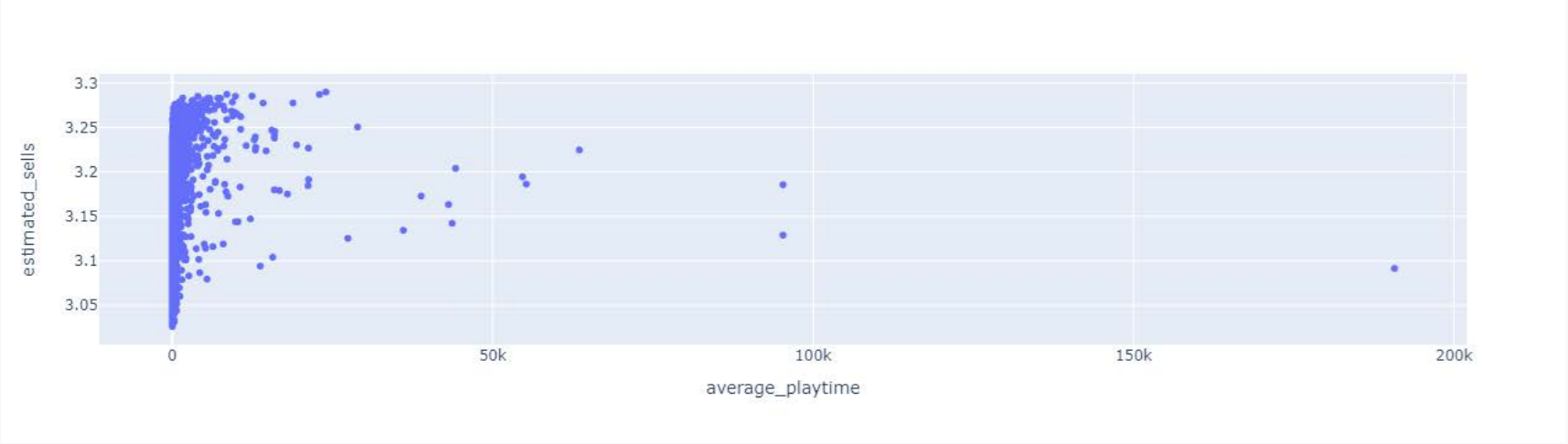
Average playtime da información de las ventas, y achievements del rating.

Scatterplots: Achievements vs ventas



Además de la concentración en cero, parece haber una reducción de ventas al aumentar logros

Scatterplots: Playtime vs ventas



Aumentos de playtime parecen generalmente aumentar ventas, salvo al ser demasiado.

Scatterplots: Precio vs ventas

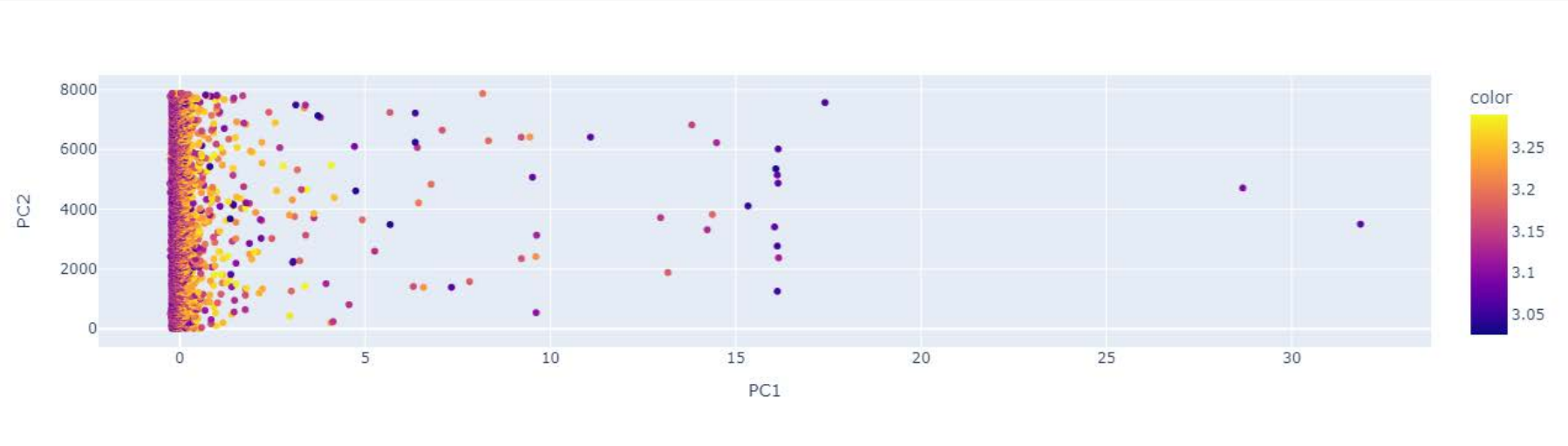
Muy ruidoso para sacar conclusiones, más que notar ciertos “clusters” de precios.

Scatterplots: Precio vs ventas

Muy ruidoso para sacar conclusiones, más que notar ciertos “clusters” de precios.

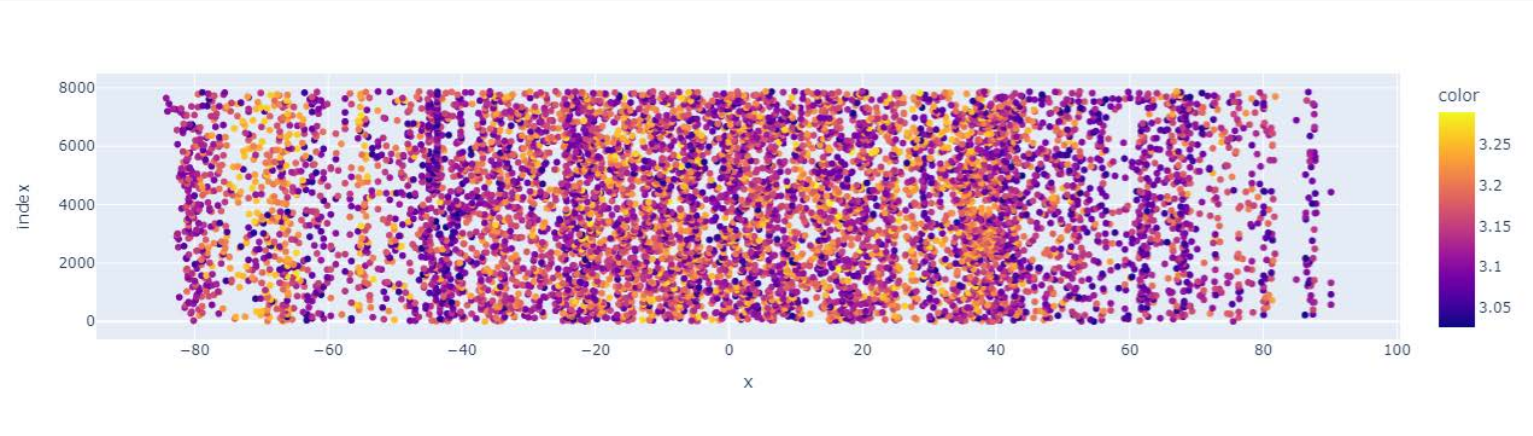
Análisis exploratorio Parte 3: Reducción de dimensionalidad

No supervisado: PCA



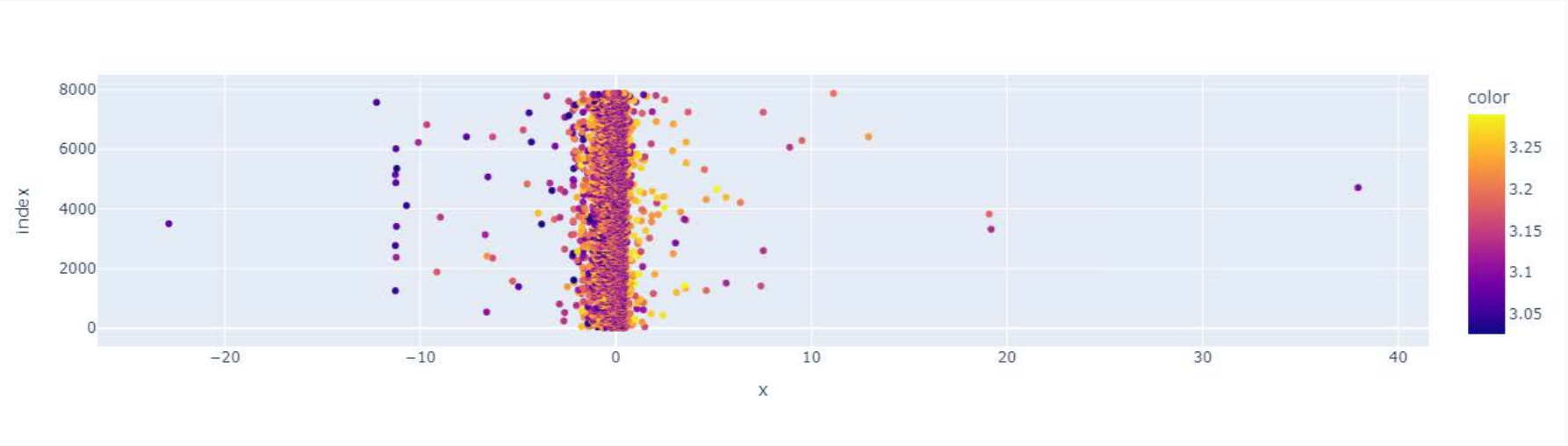
Algunos de los precios más altos se alejan acorde a la primera componente principal.

No supervisado: TSNE



Parecería que colores similares se agrupan en bandas verticales, pero no se determina una razón por que.

Supervisado: Partial Least Squares



Este método busca una proyección lineal que en vez de maximizar varianza, maximice covarianza con el target. Notamos que hay una banda concentrada de colores que tienen el precio medio, y el resto se dispersa algo más

Preparación de datos

Atributos textuales

```
bow = CountVectorizer(binary=True, tokenizer=lambda x: x.split(';'))
pd.DataFrame(bow.fit_transform(df_test.categories).todense(),
             columns=bow.get_feature_names_out())
```

Last executed at 2022-12-13 22:20:08 in 47ms

	captions available	co-op	commentary available	cross-platform multiplayer	full controller support	in-app purchases	includes level editor	includes source sdk	local co-op	local multi-player
0	0	0	0	0	1	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	0	1	0	0	0
...	...	...	...	...	...	...	...	...	...	...
871	0	0	0	0	0	1	0	0	0	0
872	0	0	0	0	0	0	0	0	0	0
873	0	0	0	0	0	0	0	0	0	0
874	0	1	0	0	0	0	0	0	0	0
875	0	1	0	0	0	0	0	0	0	0

Se utiliza CountVectorizer, aprovechando que genera las nuevas features separando por ;. Esto se hace para genres, categories, platforms y tags para obtener una matriz binaria sparse

Atributos textuales

- Para reducir dimensionalidad y evitar palabras repetidas, consideré opcionalmente usar TruncatedSVD con 100 componentes
- Short description tiene el mismo tratamiento con TfidfVectorizer, pero eventualmente se vió que era redundante tener ambos: Los otros textos ya daban el mismo puntaje que agregar esto, y así evitamos ruido

Atributos numéricos

- La fecha se separa como se mencionó antes
- El resto se pasa por StandardScaler

Modelos

Baseline: Antes de buscar hiperparámetros

- Lineal: SGD, SVC
- Naïve Bayes: Distintas variantes
- Árboles: CART, ExtraTrees, XGB, LGBM, CatBoost
- KNN
- Variar si usar TruncatedSVD o nó, y qué variables de texto se usaron

Estos se probaron en primera instancia con un hold-out de 30%, y los únicos modelos que valían la pena fueron CatBoost (mejor regresor) y LGBM (mejor clasificador).

Búsqueda de hiperparámetros

- Se utiliza 10-fold CV, que sigue la proporción train-test que tenemos en realidad
- Se apoya en las guías de usuarios de las librerías para encontrar los hiperparámetros más influyentes
- Se realiza con y sín TruncatedSVD

CatBoost

- Recomienda cambiar el número de árboles (1000 por defecto) por under/over fit, por lo cual se considera 500 y 2000 además.
- Recomienda cambiar la profundidad entre 6 a 10, así que se prueban todas estas combinaciones.

TruncatedSVD	Test	Competencia
Sí	0.17	0.61
No	0.18	0.38

Ganador: 2000 árboles con profundidad máxima 7.

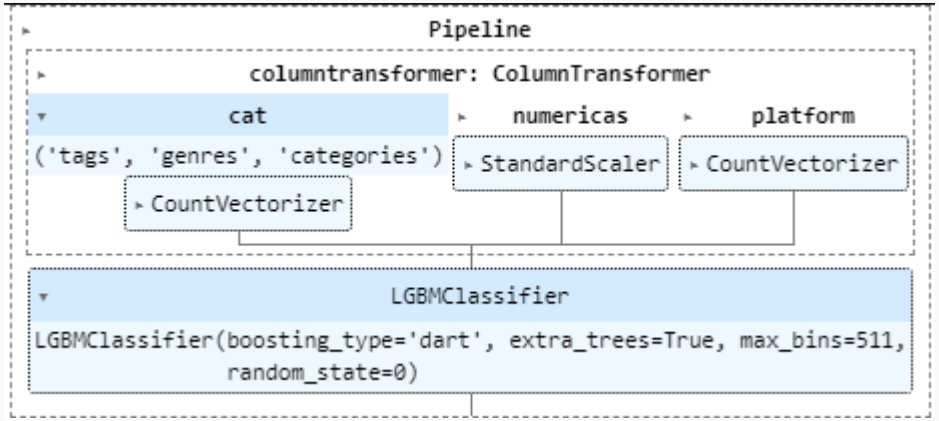
LightGBM

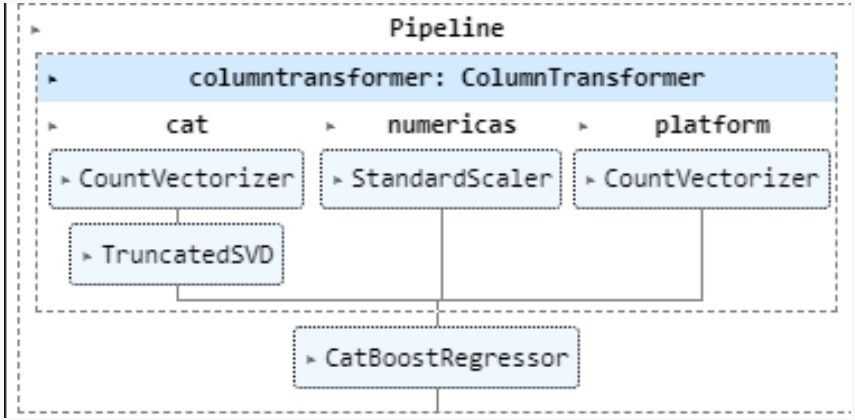
- Es mucho más veloz, así que se prueba una cantidad enorme de hiperparámetros en menos tiempo (<10 minutos, lo que me tomó lavar la loza)
- 3 tipos de boosting, si se usa extra aleatoriedad o no
- Cantidad de hojas (31, 63, 127) y max_bin (127, 255, 511) ambas recomendaban subir o bajar del por defecto

TruncatedSVD	Test	Competencia
Sí	0.33	0.35
No	0.33	0.37

Ganador: DART, Extra trees, 511 maxbins y 31 hojas máximas

Pipelines finales





Lo que no funcionó

- Cambiar los boosters usados para cada problema, y distintos modelos
- Usar búsquedas más grandes: Corrí 100.000 modelos LGBM durante la noche para mantener los mismos resultados
- Utilizar AutoML: Perdí 1 submission por formato equivocado, y la que tuvo formato correcto daba mejor puntaje en validación, pero quedó por el baseline en competencia.
- Agregar short description, con distintos vectorizadores
- Quedarse sólo con lo numérico, o sólo con lo textual

Conclusiones

- Se logró entrenar modelos de alto desempeño basados en Gradient Boosted Trees, con características obtenidas por bag-of-words binario, sin requerir técnicas más difíciles (BERT, Stacking, datos externos)

- El desempeño en competencia variaba según el uso de reducción de dimensionalidad previo, pero esto no se vió en el conjunto de testing, por lo cual un esquema de validación más cuidadoso debería ser usado para mejorar los resultados (haciendo mejores submissions).

- El problema de clasificación parece ser más difícil, y esto es intuitivo, dado que no todas las clases son igualmente discriminables por su naturaleza ordinal

¡Muchas gracias por la atención!