Predicción de temáticas de Lego

Aprendizaje de máquina 1 - CEIA (UBA)

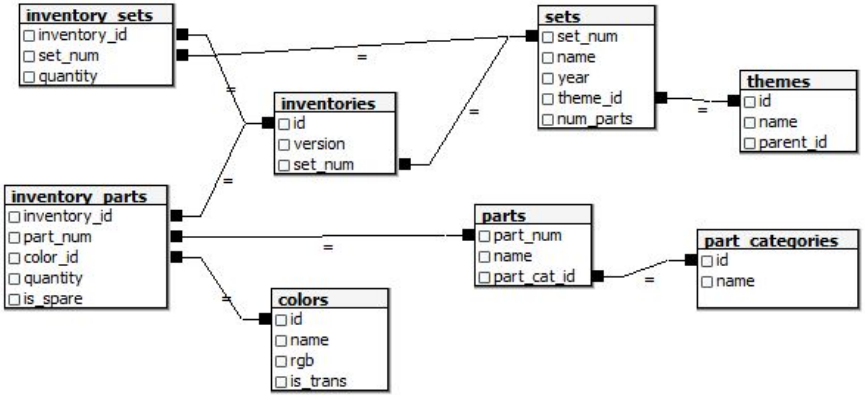
Mariana Taglio

Índice

Objetivo

Predecir la temática de un set de Lego en base a sus partes

Obtención del dataset



https://www.kaggle.com/datasets/rtatman/lego-database

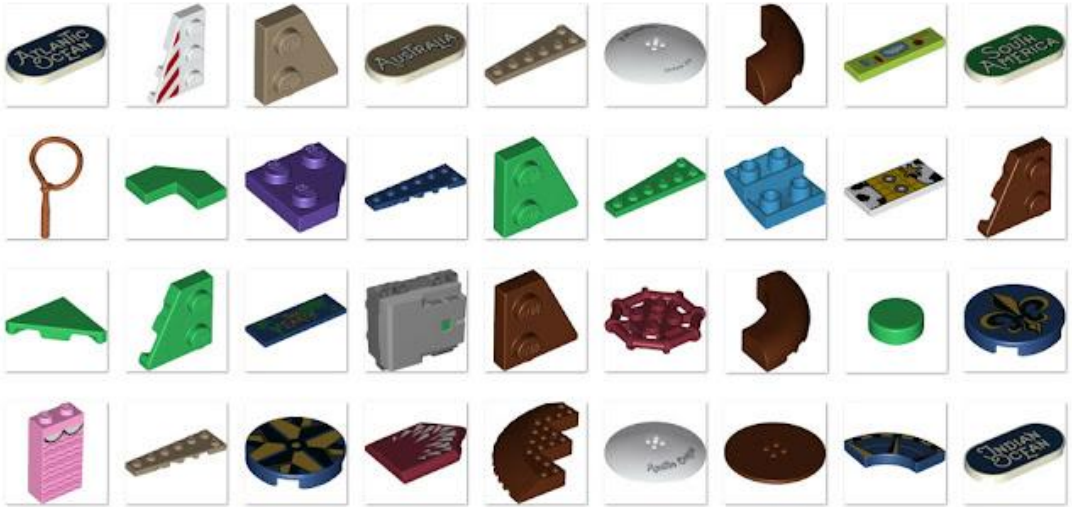
Esquema de 8 tablas

1) renaming de columnas
2) inner join de las tablas
3) concatenación y delete de columnas duplicadas
4) drop de 182 nulos
5) dataframe final con 580069 rows y 18 columnas

Entendimiento de los datos

set_num	00-1	object
set_name	Weetabix Castle	object
year	1970	int64
theme_id	414	int64
num_parts	471	int64
theme_name	Castle	object
parent_id	411.0	float64
inv_id	5574	int64
inv_version	1	int64
part_num	29c01	object
color_id	4	int64
quantity	8	int64
is_spare	f	object
color_name	Red	object
rgb	C91A09	object
is_trans	f	object
part_name	Window 1 x 1 x 2 with Glass	object
part_cat_id	16.0	float64

Entendimiento de los datos - Parts



Entendimiento de los datos - Inventories

Entendimiento de los datos - Sets

Theme : Star Wars

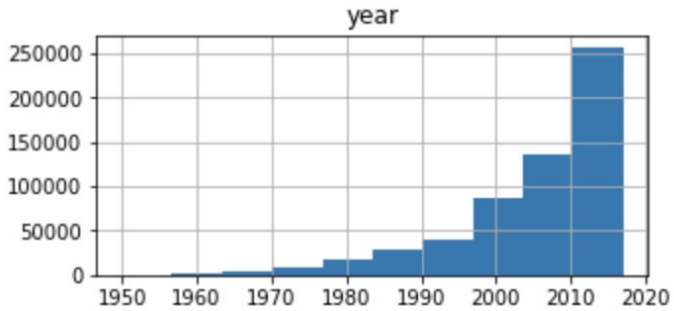
Análisis exploratorio

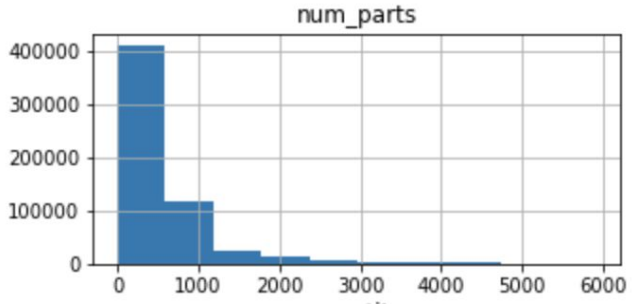
Variables de entrada y salida

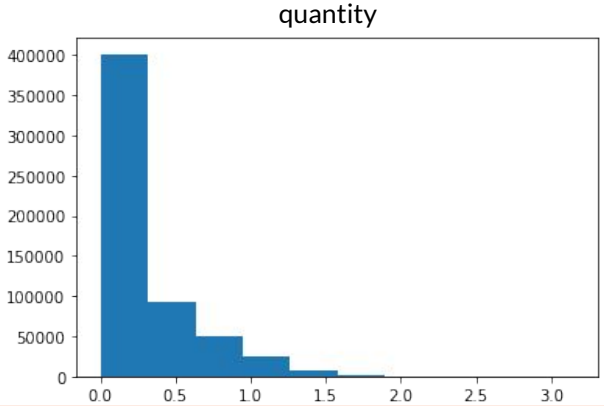
set_num	00-1	object
set_name	Weetabix Castle	object
year	1970	int64
theme_id	414	int64
num_parts	471	int64
theme_name	Castle	object
parent_id	411.0	float64
inv_id	5574	int64
inv_version	1	int64
part_num	29c01	object
color_id	4	int64
quantity	8	int64
is_spare	f	object
color_name	Red	object
rgb	C91A09	object
is_trans	f	object
part_name	Window 1 x 1 x 2 with Glass	object
part_cat_id	16.0	float64

- El objetivo es predecir la temática de un set. Usaremos la variable parent_id (Unique ID for the larger theme, if there is one.)
- La predicción se hará en base al contenido del set, por lo que las variables de entrada que nos servirán para este problema de clasificación múltiple son:
  - year, quantity (numéricas)
  - inv_id, inv_version, is_spare, color_id, part_cat_id(categóricas).
- Part_num es una variable de una cardinalidad muy alta con el doble de columnas que el total de rows. Para ello decidimos utilizar part_cat_id que tiene una cardinalidad más baja.

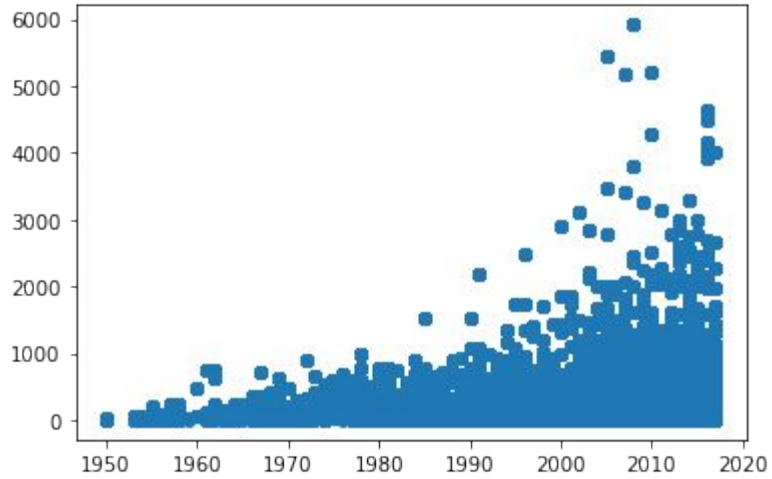
Distribución de variables numéricass







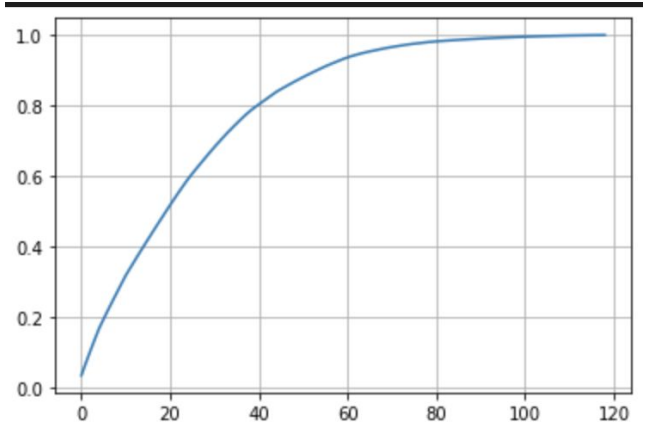
Distribución de los datos



Evolución en el tiempo del número de piezas de cada set

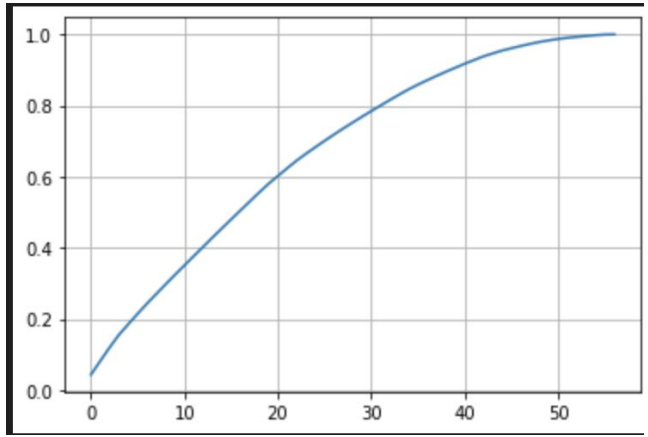
Distribución acumulada de variables categóricas

Color_id



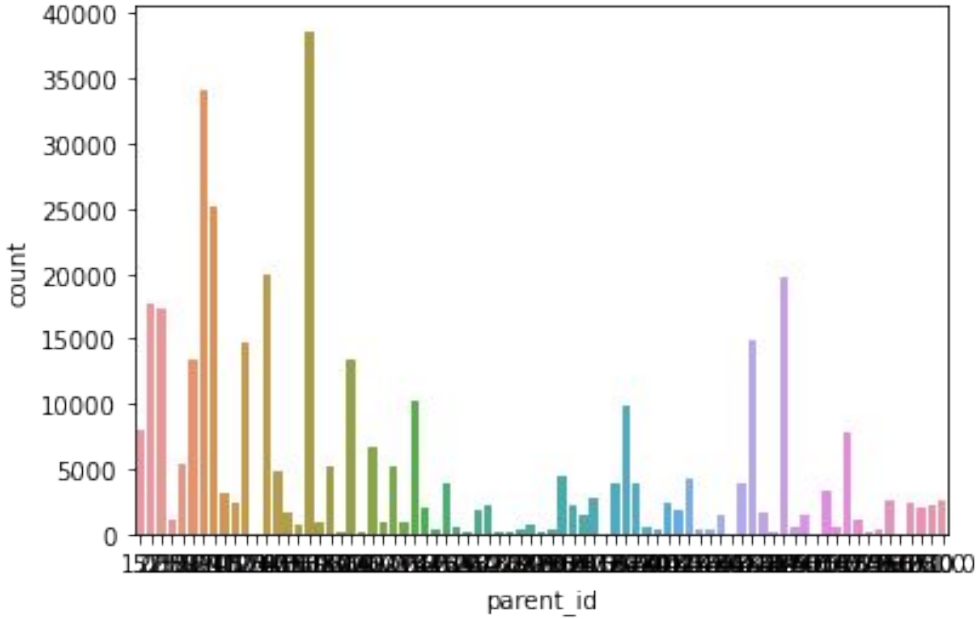
El 90% de los themes está representado por aprox 60 colores.

Part_cat_id



El 90% de los themes está representado por aprox 40 part_cat_id.

Distribución de variable de salida



- 77 unique values
- 207687 nulos

Preparación de datos

01. Imputación de nulos con most_frequent value *
02. Encoding: bag of words de color_id, is_spare y part_cat_id
03. Creación de matriz *X*: groupby set_num y agregamos variables de entrada, y mergeamos los bag of words.
04. Quitamos la variable target y la guardamos en *y*
05. Train, test split

Bag of words

El método consiste en contar la ocurrencias de estas variables en cada set.

¿Cómo?

Agrupamos por set_num y la variable categórica en cuestión (por ejemplo color_id), sumamos las cantidades que hay en cada inventario y luego creamos una pivot table

Pivot

df

	foo	bar	baz	zoo
0	one	A	1	x
1	one	B	2	y
2	one	C	3	z
3	two	A	4	q
4	two	B	5	w
5	two	C	6	t

```
df.pivot(index='foo',
         columns='bar',
         values='baz')
```

bar	A	B	C
foo			
one	1	2	3
two	4	5	6

	color_id1	color_id2	color_id3
set_num			
00-1	2	0	3
tf05-1	0	4	1

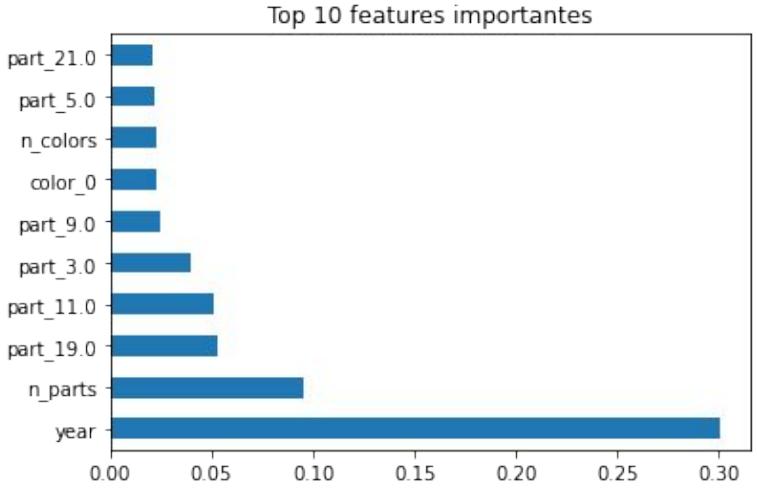
Entrenamiento

Modelos

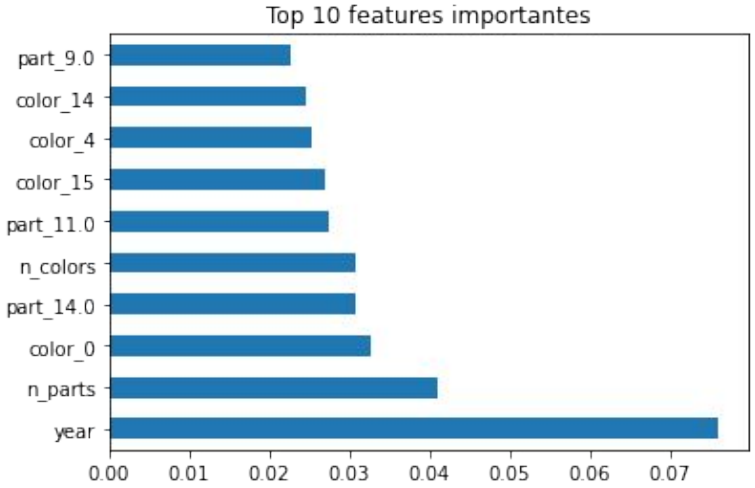
1. Baseline model: Logistic Regression (max_iter=100)
2. Decision tree classifier (max_depth=20, criterion=”entropy”)
3. Random forest (n_estimators=150)

Métricas de evaluación: accuracy, precision, recall y f1

Feature Importance



Decision tree



Random forest

Resultados

	Accuracy	Precision	Recall	F1Score
LogisticRegression	0.39	0.24	0.39	0.25
Decision Tree - md=10	0.63	0.63	0.63	0.62
Decision Tree - md=20	0.66	0.66	0.66	0.66
Random Forest -md =10	0.57	0.59	0.57	0.48
Random Forest - md=None	**0.75**	**0.75**	**0.75**	**0.71**

average= “weighted”

Balanceo de clases

Métodos

1. Oversampling
2. Subsample del dataframe

Oversampling: 9x

	Accuracy	Precision	Recall	F1Score
LogisticRegression	0.17	0.47	0.19	0.19
Decision Tree -md=20	0.65	0.67	0.65	0.65
Decision Tree -md=10	0.42	0.59	0.42	0.43
Random Forest -md=10	0.42	0.39	0.42	0.42
Random Forest	**0.79**	**0.79**	**0.79**	**0.77**

Subsample = parent_id.values > 300

	Accuracy	Precision	Recall	F1Score
Decision Tree	0.66	0.67	0.66	0.65
Random Forest	**0.83**	**0.83**	**0.83**	**0.83**

Train set (5038, 178)

Train set original: (7499, 182)

Key takeaways

- **Objetivo**: predecir temática de un set de Lego
- **Variable de salida**: parent_id
- **Encoding variables categóricas**: bag of words
- **Imputación missing values**: most_frequent
- **Pain points**: clases desbalanceadas
- **Técnicas**: oversampling, kbest
- **Modelos**: lr, decistion trees, random_forest

Mejoras futuras

1. Probar otros modelos como xgboost o lightgbm
2. Probar otras técnicas de imputación, ej: datawig
3. Stratified Kfolds