Clasificador de Setas.

Modelado Machine Learning categórico o de reconocimiento de imágen.

Hugo Martín-Santos
Alumno Bootcamp Data Science
The Bridge

Lasarte-Oria/Madrid
01/XII/2023

Plan de ataque

- Obtención de Data.
- Análisis de Data.
- Diseño de modelado.
- Entrenamiento de modelos.
- Desarrollo APP.

Illustrations by Pixeltrue on icons8

Problema de Clasificación multicategórico

Retos a tener en cuenta

> La presencia de múltiples columnas con data categórica en el DataFrame planteará el reto de cómo afrontar el correcto tratamiento y pre-procesamiento.

01

Análisis

Valoración de parámetros. Descartes y primeros tratamientos.

02

Expansión de categorías

La multi-categoría se aborda creando filas por cada combinación existente.

03

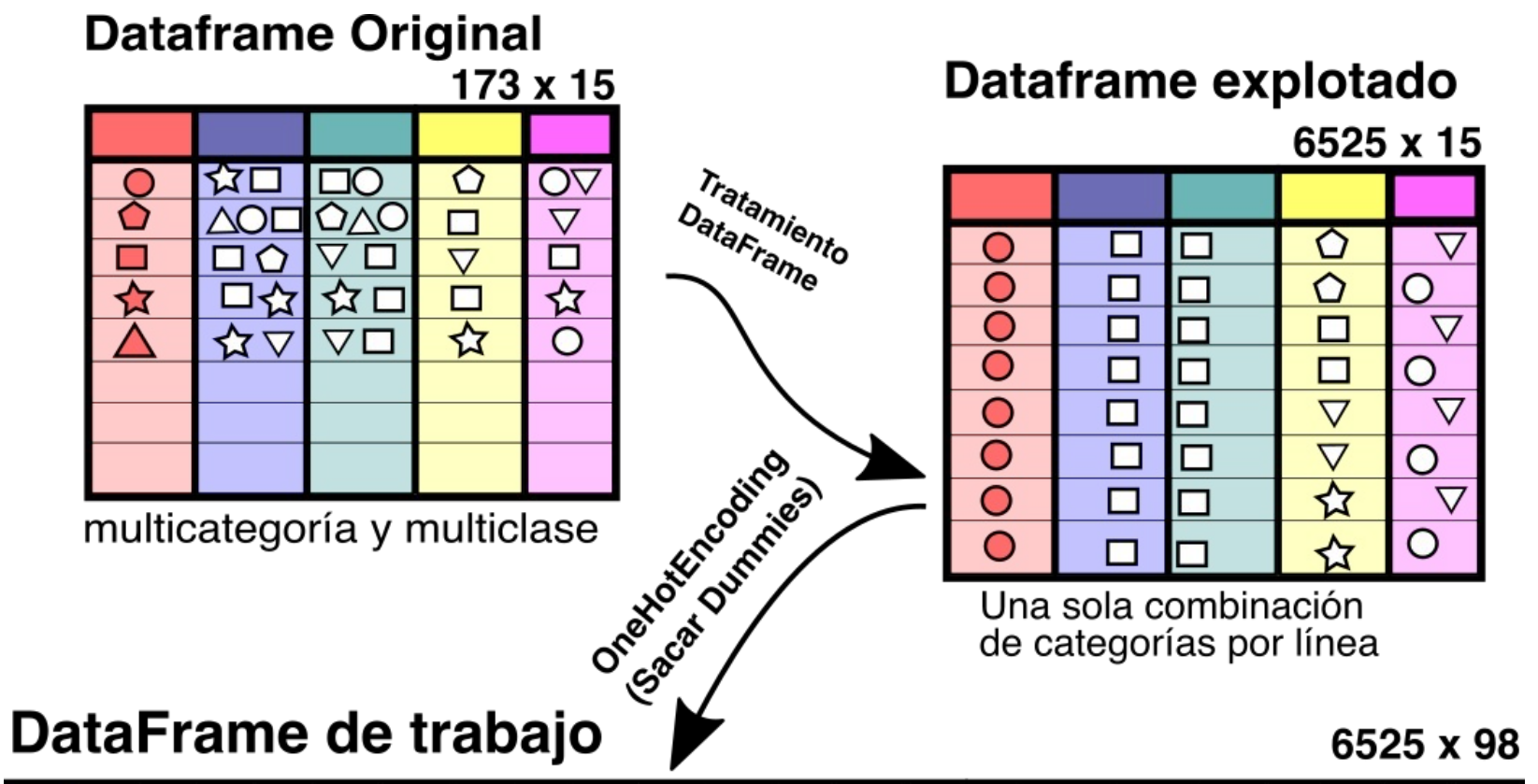
OneHotEncoding

Paso final. Pérdida de legibilidad, pero adecuación necesaria al problema de ML.

04

Modelado

Trabajo de hiperparametrización y planteamiento de modelos múltiples.


Dataframe Original
173 x 15
multicategoría y multiclase
Tratamiento DataFrame
Dataframe explotado
6525 x 15
Una sola combinación de categorías por línea
OneHotEncoding (Sacar Dummies)
DataFrame de trabajo
6525 x 98
Columnas categóricas binarias, adecuadas para el aprendizaje ML

Original

	name	cap-diameter	cap-shape	cap-surface	cap-color	gill-color	stem-height	stem-width	stem-surface	stem-color	veil-color	has-ring	ring-type	habitat	season
0	Fly Agaric	[10, 20]	[x, f]	[g, h]	[e, o]	[w]	[15, 20]	[15, 20]	[y]	[w]	[w]	[t]	[g, p]	[d]	[u, a, w]
1	Panther Cap	[5, 10]	[p, x]	[g]	[n]	[w]	[6, 10]	[10, 20]	[y]	[w]	[w]	[t]	[p]	[d]	[u, a]
2	False Panther Cap	[10, 15]	[x, f]	NaN	[g, n]	[w]	[10, 12]	[10, 20]	NaN	[w]	[w]	[t]	[e, g]	[d]	[u, a]
3	The Blusher	[5, 15]	[x, f]	NaN	[n]	[w]	[7, 15]	[10, 25]	NaN	[w]	[w]	[t]	[g]	[d]	[u, a]
4	Death Cap	[5, 12]	[x, f]	[h]	[r]	[w]	[10, 12]	[10, 20]	NaN	[w]	[w]	[t]	[g, p]	[d]	[u, a]
5	False Death Cap	[4, 9]	[x]	NaN	[w, y]	[w]	[5, 7]	[10, 15]	NaN	[w, y]	[y, w]	[t]	[g]	[d]	[u, a]
6	Destroying Angel	[5, 10]	[b]	[t]	[w]	[w]	[10, 15]	[10, 15]	[y]	[w]	[w]	[t]	[l, e]	[d]	[u, a]
7	Tawny Grisette	[4, 8]	[c, x]	[h, t]	[n]	[w]	[10, 15]	[10, 15]	[s]	[w, n]	[w]	[f]	[f]	[d]	[u, a]
8	Parasol Mushroom	[10, 25]	[p, f]	[y]	[w, n]	[w]	[15, 35]	[15, 25]	NaN	[n]	NaN	[t]	[m]	[m, d]	[u, a]
9	Shaggy Parasol	[12, 18]	[x]	[e, y]	[n]	[w]	[8, 12]	[15, 20]	NaN	[w]	NaN	[t]	NaN	[g, d]	[u, a]

Trabajada

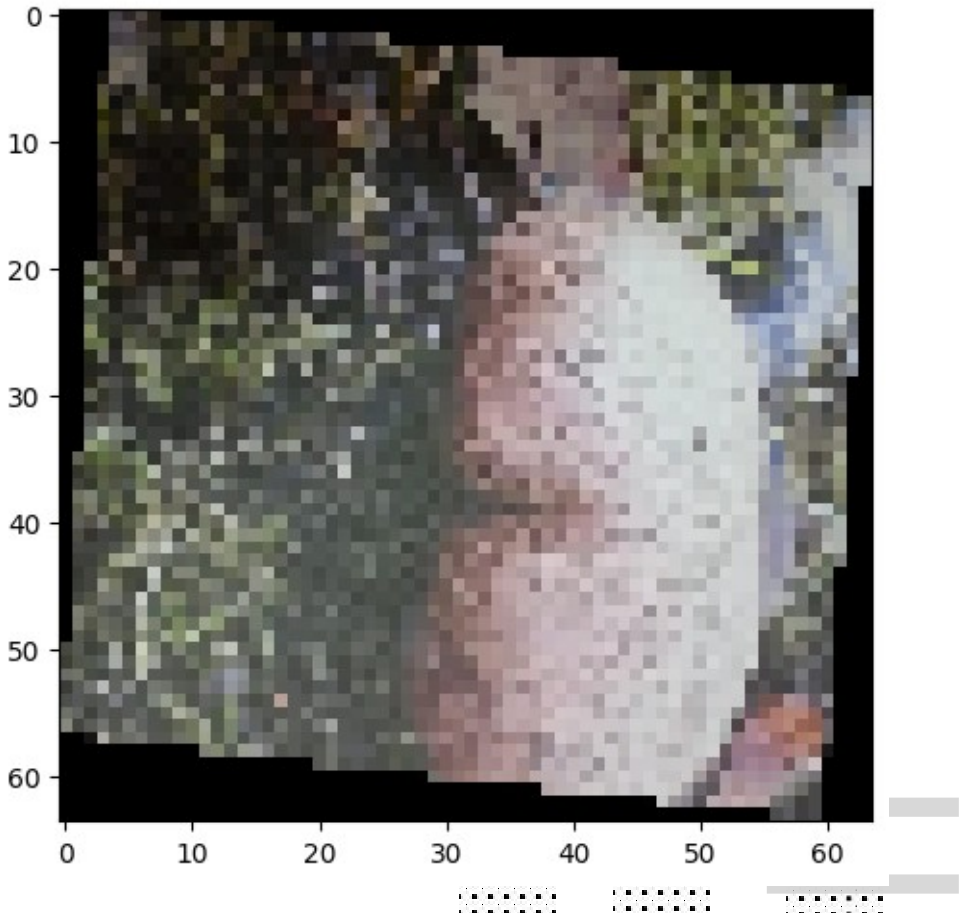
	name	cap-diameter	stem-height	stem-width	has-ring	cap-shape_b	cap-shape_c	cap-shape_f	cap-shape_o	cap-shape_p	...	habitat_h	habitat_l	habitat_m	habitat_p	habitat_u	habitat_w	season_a	season_s	season_u	season_w
0	Fly Agaric	15.0	17.5	17.5	1	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
1	Fly Agaric	15.0	17.5	17.5	1	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
2	Fly Agaric	15.0	17.5	17.5	1	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
3	Fly Agaric	15.0	17.5	17.5	1	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
4	Fly Agaric	15.0	17.5	17.5	1	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
5	Fly Agaric	15.0	17.5	17.5	1	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
6	Fly Agaric	15.0	17.5	17.5	1	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
7	Fly Agaric	15.0	17.5	17.5	1	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
8	Fly Agaric	15.0	17.5	17.5	1	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Reconocimiento de imágenes
Red Neuronal Convolucional

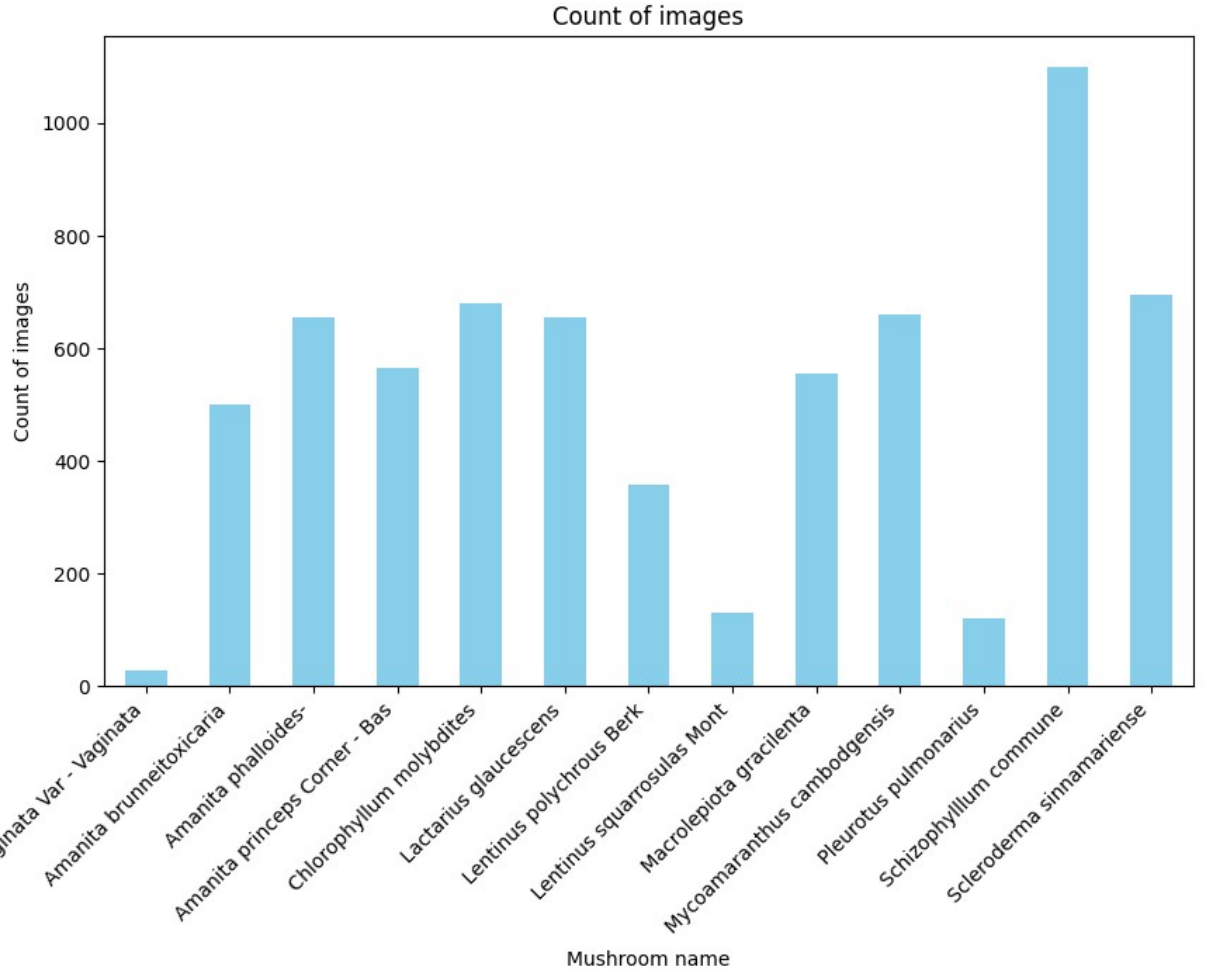
Retos a tener en cuenta

> El preprocesamiento de imágenes y el diseño de una estructura del modelo apta para su procesamiento será vital para conseguir el éxito.

Procesamiento de imágenes



Cantidad de imágenes por target.



Este desequilibrio en la representación de ciertas categorías habrá que tratarlo para evitar el overfitting/under sampling.

Diseño de Estructura neuronal modelo tensorflow Keras

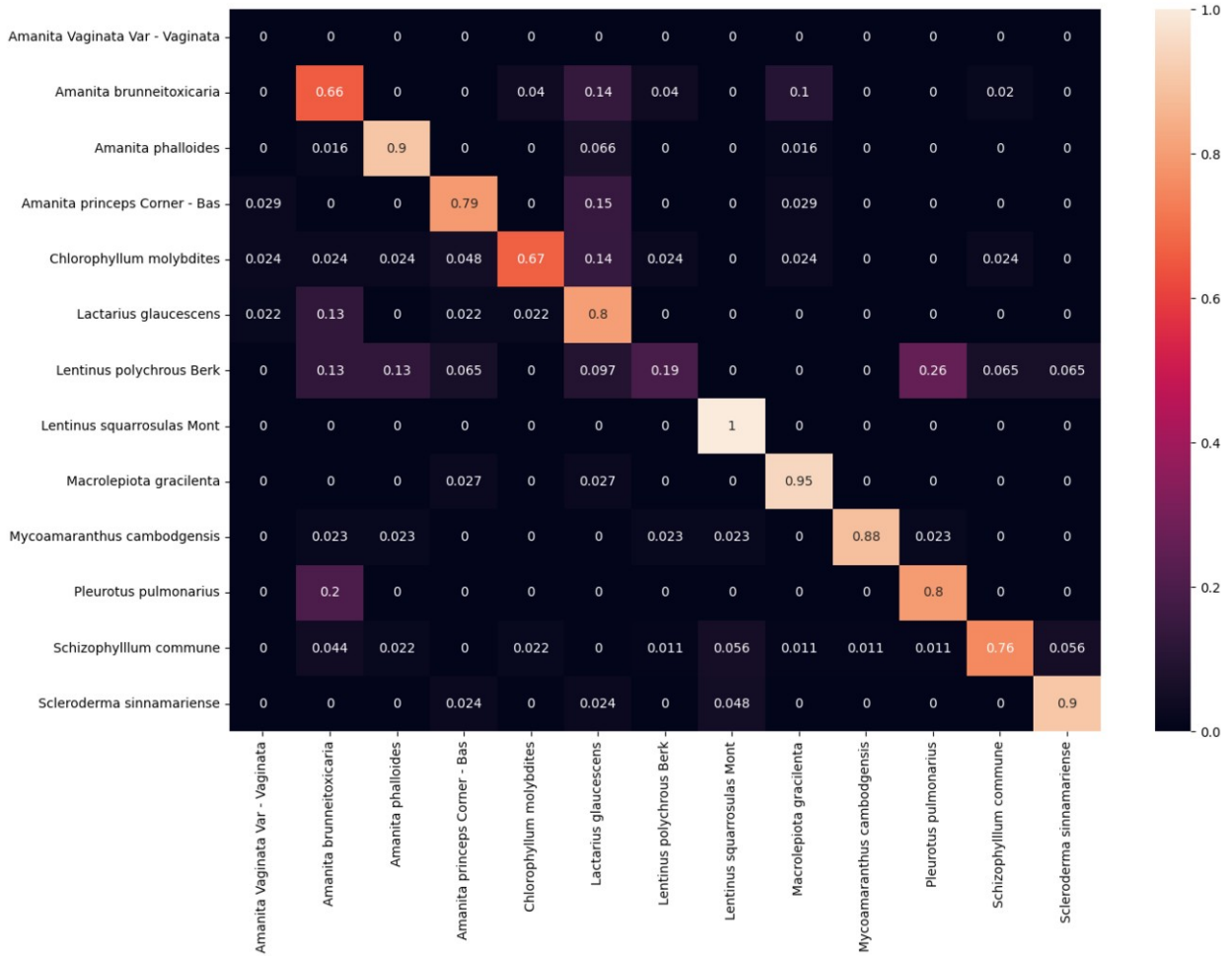
Capas Red Neuronal

```python
layers = [
    keras.layers.Conv2D(128, (3,3), activation='relu', input_shape=(48,48,3)),
    keras.layers.MaxPooling2D(pool_size=(2,2)),
    keras.layers.Dropout(0.1),

    keras.layers.Conv2D(64, (3,3), activation='relu'),
    keras.layers.MaxPooling2D(pool_size=(2,2)),
    keras.layers.Dropout(0.1),

    keras.layers.Conv2D(64, (3,3), activation='relu'),
    keras.layers.MaxPooling2D(pool_size=(2,2)),
    keras.layers.Dropout(0.1),

    keras.layers.Flatten(),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(32, activation='relu'),
    keras.layers.Dropout(0.1),
    keras.layers.Dense(13, activation='sigmoid')
]

cnn_model = keras.Sequential(layers)

cnn_model.compile(optimizer = 'adam',
                  loss = 'categorical_crossentropy',
                  metrics = ['accuracy'])
```

Sumario estructura

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_7 (Conv2D)           (None, 46, 46, 128)       3584

 max_pooling2d_7 (MaxPoolin  (None, 23, 23, 128)       0
 g2D)

 dropout (Dropout)           (None, 23, 23, 128)       0

 conv2d_8 (Conv2D)           (None, 21, 21, 64)        73792

 max_pooling2d_8 (MaxPoolin  (None, 10, 10, 64)        0
 g2D)

 dropout_1 (Dropout)         (None, 10, 10, 64)        0

 conv2d_9 (Conv2D)           (None, 8, 8, 32)          18464

 max_pooling2d_9 (MaxPoolin  (None, 4, 4, 32)          0
 g2D)

 dropout_2 (Dropout)         (None, 4, 4, 32)          0

...
Total params: 163181 (637.43 KB)
Trainable params: 163181 (637.43 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Matriz de confusión
Test nunca antes visto por el modelo



Ciertas categorías no terminan de ser bien asimiladas por el modelo. Quedaría profundizar en el porqué de los casos particularmente malos.

En general el modelo se comporta según lo esperado.