

Proyecto de TFM

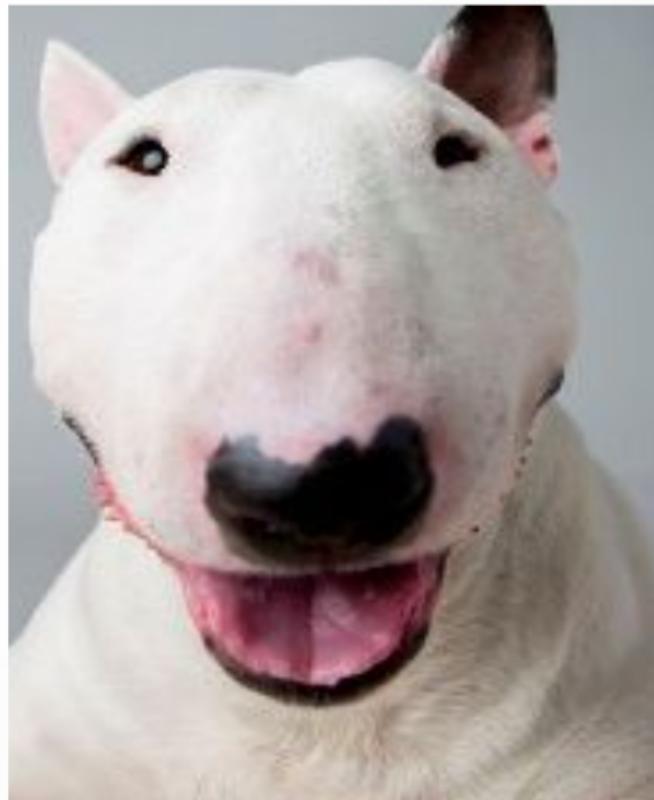
P.E.T.

Pet enhancement Transition

Empezar

ÍNDICE

1. Justificación y descripción del proyecto
2. Obtención de datos
3. Descripción y limpieza de los datos
4. Exploración y visualización de los datos
5. Preparación de los datos para los algoritmos de Machine Learning
6. Entrenamiento del modelo y comprobación del rendimiento
7. Procesamiento de Lenguaje Natural
8. Aplicación web
9. Autores
- 10 Gracias



JUSTIFICACIÓN Y DESCRIPCIÓN DEL PROYECTO

- **Predicción:** Razas de perros y gatos en imágenes
- **Datasets:** Kaggle
- **Modelos:** VGG19, RestNet50 e InceptionV3
- **Página web:** Streamlit
- **Apartados importantes:** Predictor y Adoptar un animal
- **NLP:** Traducción de textos
- **Big Data:** MongoDB
- **Objetivo:** Facilitar la identificación y fomentar la adopción



OBTENCIÓN DE DATOS



Cats:

- 15 razas de gatos.
- 375 imágenes por raza.
- Se deberá tratar para mezclarla con el dataset de perros.
- 10 para test, 10 para valid y el resto para train.

Dogs:

- 70 razas.
- Imágenes separadas en train, valid y test.
- Entre 65 y 198 imágenes por raza para entrenar.
- 10 imágenes para la validación y 10 para test.





DESCRIPCIÓN Y LIMPIEZA DE LOS DATOS

	path	breed	type	dataset
0	Dog/test/Great Perenees/10.jpg	Great Perenees	Dog	test
1	Dog/test/Great Perenees/09.jpg	Great Perenees	Dog	test
2	Dog/test/Great Perenees/01.jpg	Great Perenees	Dog	test
3	Dog/test/Great Perenees/03.jpg	Great Perenees	Dog	test
4	Dog/test/Great Perenees/08.jpg	Great Perenees	Dog	test
...
9341	Dog/train/Shiba Inu/32.jpg	Shiba Inu	Dog	train
9342	Dog/train/Shiba Inu/46.jpg	Shiba Inu	Dog	train
9343	Dog/train/Shiba Inu/14.jpg	Shiba Inu	Dog	train
9344	Dog/train/Shiba Inu/40.jpg	Shiba Inu	Dog	train
9345	Dog/train/Shiba Inu/84.jpg	Shiba Inu	Dog	train

9346 rows × 4 columns

DataFrame Dogs

- **path:** Ruta de la imagen.
- **breed:** Raza de mascota.
- **type:** Tipo de mascota.
- **dataset:** Uso de los datos

+ info

DataFrame Cats

	path	breed	type	dataset
0	Cat/Birman/Birman-36689172_1935.jpg	Birman	Cat	test
1	Cat/Birman/Birman_20.jpg	Birman	Cat	test
2	Cat/Birman/Birman-34882234_2339.jpg	Birman	Cat	test
3	Cat/Birman/Birman-33559605_264.jpg	Birman	Cat	test
4	Cat/Birman/Birman_77.jpg	Birman	Cat	test
...
5620	Cat/Bombay/Bombay-18376775_8377.jpg	Bombay	Cat	train
5621	Cat/Bombay/Bombay-20394160_7978.jpg	Bombay	Cat	train
5622	Cat/Bombay/Bombay-25252179_6811.jpg	Bombay	Cat	train
5623	Cat/Bombay/Bombay_171.jpg	Bombay	Cat	train
5624	Cat/Bombay/Bombay_111.jpg	Bombay	Cat	train

5625 rows × 4 columns

dogs.info()

Visualización de nulos

+ info

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9346 entries, 0 to 9345
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   path        9346 non-null    object 
 1   breed       9346 non-null    object 
 2   type        9346 non-null    object 
 3   dataset     9346 non-null    object 
dtypes: object(4)
memory usage: 292.2+ KB
```

```
cats.info()  
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 5625 entries, 0 to 5624  
Data columns (total 4 columns):  
 #   Column      Non-Null Count  Dtype     
 ---  --          -----            
 0   path        5625 non-null    object    
 1   breed       5625 non-null    object    
 2   type        5625 non-null    object    
 3   dataset     5625 non-null    object    
 dtypes: object(4)  
 memory usage: 175.9+ KB
```

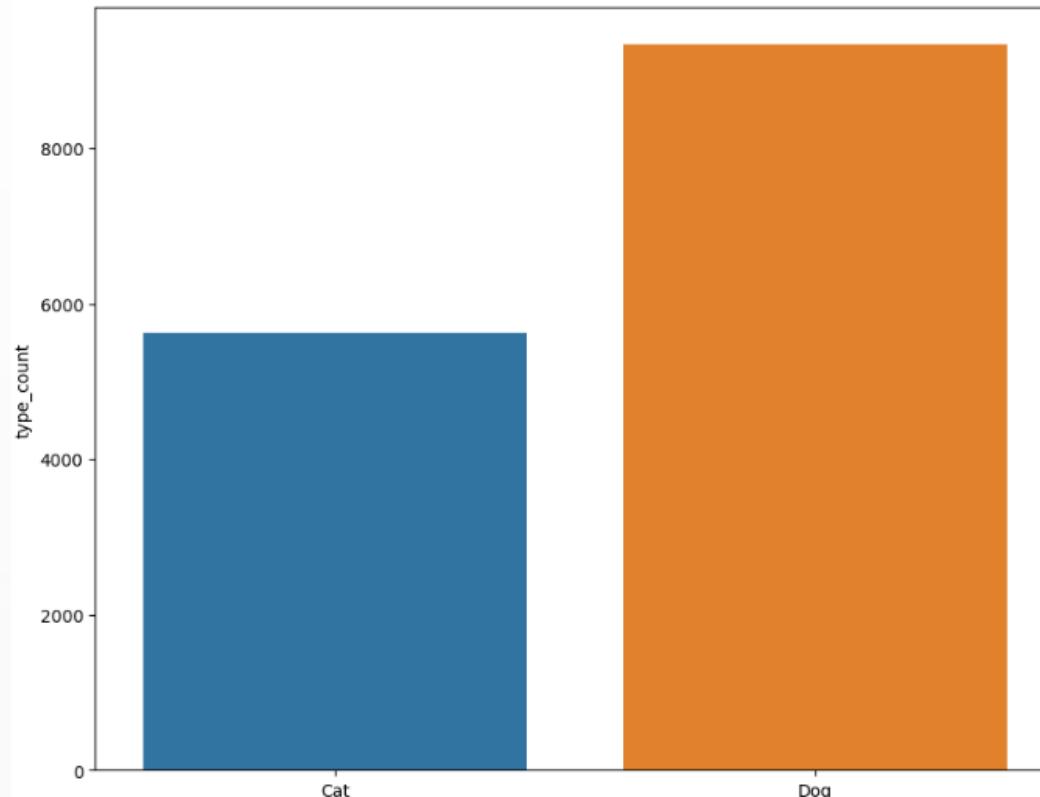


EXPLORACIÓN Y VISUALIZACIÓN DE LOS DATOS

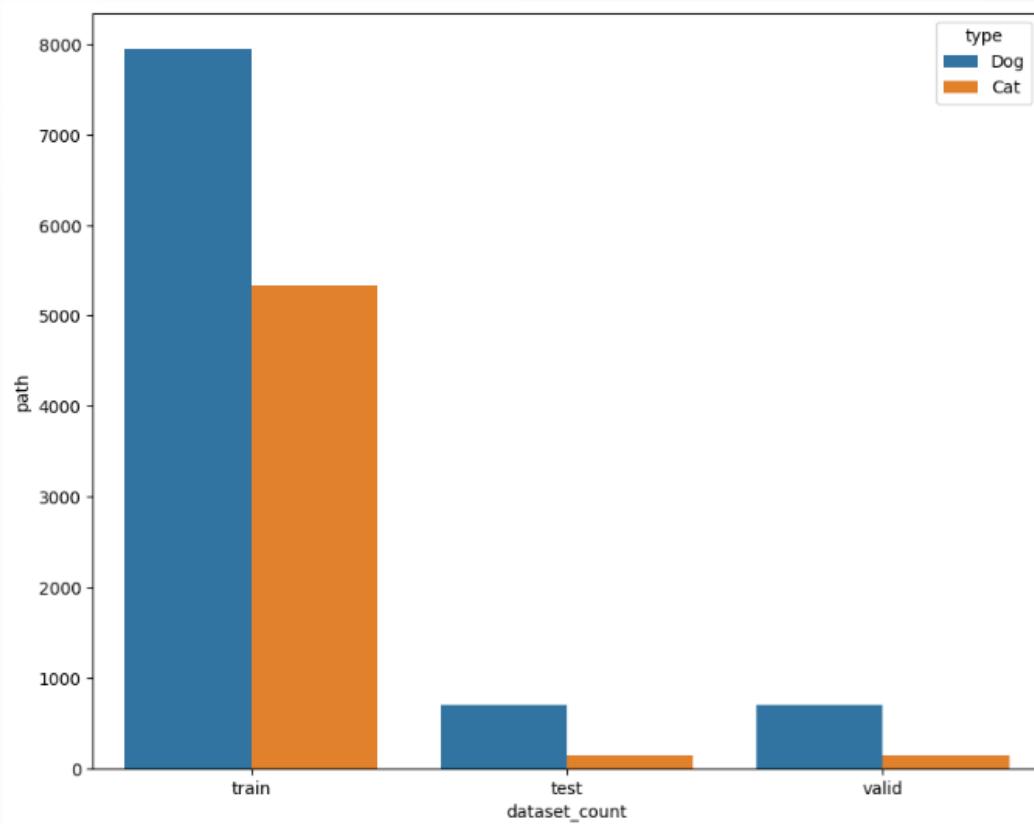
DataFrame Pets

		path	breed	type	dataset
0		Dog/test/Great Perenees/10.jpg	Great Perenees	Dog	test
1		Dog/test/Great Perenees/09.jpg	Great Perenees	Dog	test
2		Dog/test/Great Perenees/01.jpg	Great Perenees	Dog	test
3		Dog/test/Great Perenees/03.jpg	Great Perenees	Dog	test
4		Dog/test/Great Perenees/08.jpg	Great Perenees	Dog	test
...	
5620	Cat/Bombay/Bombay-18376775_8377.jpg		Bombay	Cat	train
5621	Cat/Bombay/Bombay-20394160_7978.jpg		Bombay	Cat	train
5622	Cat/Bombay/Bombay-25252179_6811.jpg		Bombay	Cat	train
5623	Cat/Bombay/Bombay_171.jpg		Bombay	Cat	train
5624	Cat/Bombay/Bombay_111.jpg		Bombay	Cat	train
14971 rows × 4 columns					

Gatos y perros



Uso de datos





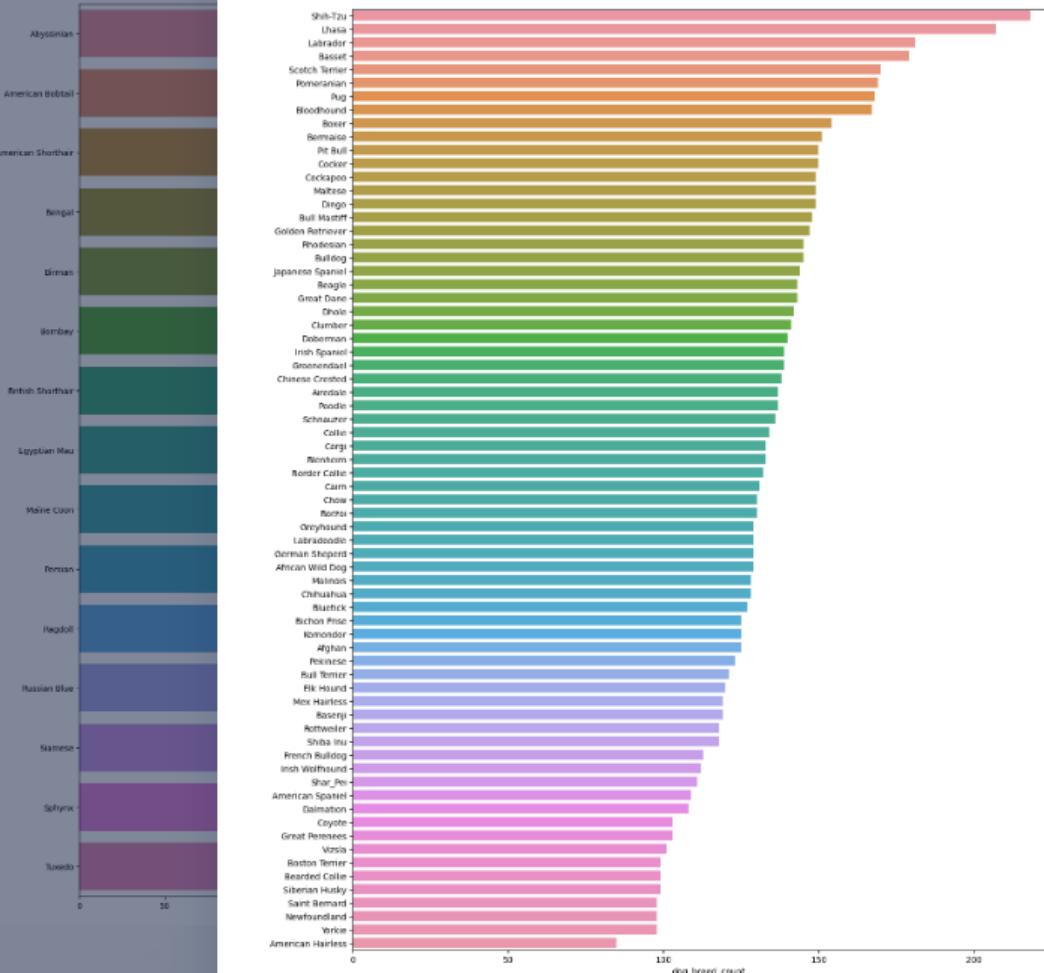
Raza de mascota

+ info

X

mascota

D



Visualización de imágenes



+ info



-

Visualización de





PREPARACIÓN
DE LOS DATOS
PARA LOS
ALGORITMOS DE
MACHINE
LEARNING

```
img_gen = ImageDataGenerator(rescale=1./255.)  
  
pets_b_uniq = pets['breed'].unique().tolist()  
  
pets_train = pets[(pets['dataset'] == 'train')]  
pets_val = pets[(pets['dataset'] == 'valid')]  
pets_test = pets[(pets['dataset'] == 'test')]  
  
pets_gen_train = img_gen.flow_from_dataframe(  
    dataframe=pets_train,  
    directory=path,  
    x_col='path',  
    y_col='breed',  
    classes=pets_b_uniq,  
    shuffle=True,  
    class_mode='categorical',  
    target_size=(256, 256),  
    batch_size=64  
)
```

- Generador
- Lista de clases
- Agrupación de datos
- Variables:
 - pet_gen_train
 - pet_gen_valid
 - pet_gen_test
- Parámetros que cambian:
dataframe, batch_size y shuffle

+ info

```
img_gen = ImageDataGenerator(rescale=1./255.)  
  
pets_b_uniq = pets['breed'].unique().tolist()  
  
pets_train = pets[(pets['dataset'] == 'train')]  
pets_val = pets[(pets['dataset'] == 'valid')]  
pets_test = pets[(pets['dataset'] == 'test')]  
  
pets_...  
Found 13271 validated image filenames belonging to 85 classes.  
Found 850 validated image filenames belonging to 85 classes.  
Found 850 validated image filenames belonging to 85 classes.  
  
class_mode='categorical',  
target_size=(256, 256),  
batch_size=64  
)
```

- Generador
- Lista de clases
- Agrupación de datos



+ info



ENTRENAMIENTO
DEL MODELO Y
COMPROBACIÓN
DEL
RENDIMIENTO

CARGA DEL MODELO

+ info

```
v_model = VGG19(weights='imagenet', include_top=False, input_shape=(256, 256, 3))  
v_model.trainable = False  
v_model.summary()
```

CREACIÓN DE CAPAS NO CONVOLUCIONALES

+ info

```
vgg = Sequential()  
vgg.add(v_model)  
  
vgg.add(GlobalAveragePooling2D())  
vgg.add(Dense(128))  
vgg.add(Dropout(0.2))  
vgg.add(Dense(len(pets_b_uniq), activation='softmax'))  
  
vgg.build([None, 256, 256, 3])  
vgg.summary()
```

CARGA DEL MODELO

+ info

```
v_model = VGG19(weights='imagenet', include_top=False, input_shape=(256, 256, 3))  
v_model.trainable = False
```



```
r_model = ResNet50(weights='imagenet', include_top=False, input_shape=(256, 256, 3))  
r_model.trainable = False  
r_model.summary()
```

```
i_model = InceptionV3(weights='imagenet', include_top=False, input_shape=(256, 256, 3))  
i_model.trainable = False  
i_model.summary()
```

```
vgg.build([None, 256, 256, 3])  
vgg.summary()
```

VGG 19

```
Model: "sequential"
Layer (type)          Output Shape       Param #
=================================================================
vgg19 (Functional)    (None, 8, 8, 512)   20024384
global_average_pooling2d (Globa
lAveragePooling2D)
dense (Dense)         (None, 128)        65664
dropout (Dropout)     (None, 128)        0
dense_1 (Dense)       (None, 85)         10965
=====
Total params: 20,101,013
Trainable params: 76,629
Non-trainable params: 20,024,384
```

RestNet 50

```
Model: "sequential_1"
Layer (type)          Output Shape       Param #
=================================================================
resnet50 (Functional) (None, 8, 8, 2048)  23587712
global_average_pooling2d_1 (GlobalAveragePooling2D)
dense_2 (Dense)        (None, 128)        262272
dropout_1 (Dropout)    (None, 128)        0
dense_3 (Dense)        (None, 85)         10965
=====
Total params: 23,860,949
Trainable params: 273,237
Non-trainable params: 23,587,712
```

Inception V3

```
Model: "sequential_2"
Layer (type)          Output Shape       Param #
=================================================================
inception_v3 (Functional) (None, 6, 6, 2048)  21802784
global_average_pooling2d_2 (GlobalAveragePooling2D)
dense_4 (Dense)        (None, 128)        262272
dropout_2 (Dropout)    (None, 128)        0
dense_5 (Dense)        (None, 85)         10965
=====
Total params: 22,076,021
Trainable params: 273,237
Non-trainable params: 21,802,784
```

COMPILACIÓN

```
vgg.compile(optimizer=Adam(),  
            loss="categorical_crossentropy",  
            metrics=["accuracy"])
```

ENTRENAMIENTO

```
cb_early_stopper = EarlyStopping(monitor='val_loss', patience=15)  
  
/_v = vgg.fit(pets_gen_train, validation_data=pets_gen_valid, callbacks=cb_early_stopper, epochs=100)
```

EVALUACIÓN DE TEST

+ info

```
850/850 [=====] - 17s 19ms/step - loss: 1.1250 - accuracy: 0.6729  
Model Test Loss: 1.1249868869781494  
Model Test Accuracy: 0.6729411482810974
```

COMPILACIÓN

```
vgg.compile(optimizer=Adam(),
            loss="categorical_crossentropy",
            metrics=["accuracy"])
```

RestNet 50

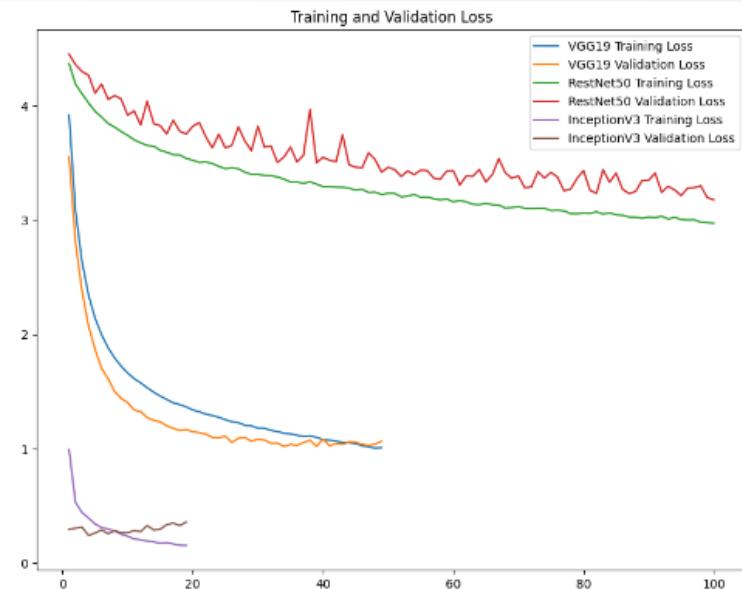
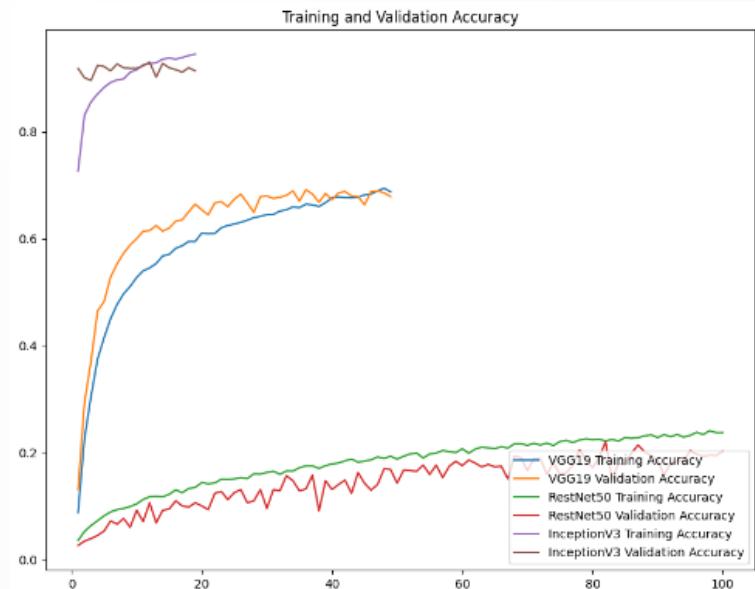
```
850/850 [=====] - 16s 17ms/step - loss: 3.2372 - accuracy: 0.1929
Model Test Loss: 3.2371644973754883
Model Test Accuracy: 0.19294117391109467
```

Inception V3

```
850/850 [=====] - 15s 17ms/step - loss: 0.3263 - accuracy: 0.9094
Model Test Loss: 0.32627758383750916
Model Test Accuracy: 0.9094117879867554
```

```
850/850 [=====] - 17s 19ms/step - loss: 1.1250 - accuracy: 0.6729
Model Test Loss: 1.1249868869781494
Model Test Accuracy: 0.6729411482810974
```

COMPROBACIÓN DEL RENDIMIENTO



GUARDADO

```
label_map = (pets_gen_train.class_indices)
inv_map = {v: k for k, v in label_map.items()}
import json

with open('breeds.json', 'w') as bjson:
    json.dump(inv_map, bjson)

incep.save('inceptionV3.h5')
```

CARGA

+ info

```
class Model:
    def __init__(self, weights_path: str, classes_name_path: str) -> None:
        # Carga del modelo y clases

        self.model = load_model(weights_path)
        with open(classes_name_path, 'r') as c:
            classes = json.load(c)
        self.classes = {int(k): v for k, v in classes.items()}

    def predict(self, img_path: str) -> str:
        # Prediccion de la imagen

        img = keras.preprocessing.image.load_img(img_path, target_size=(256, 256))
        img_array = keras.utils.img_to_array(img)
        img_batch = np.expand_dims(img_array, axis=0)
        img_preprocessed = preprocess_input(img_batch)
        pred = self.model.predict(img_preprocessed)
        pred = pred.argmax()

        return self.classes[pred]
```

CARGA

+ info

```
class Model:  
    def __init__(self, weights_path: str, classes_name_path: str) -> None:  
        # Carga del modelo y clases
```

```
model = Model(weights_path='inceptionV3.h5', classes_name_path='breeds.json')  
  
import os  
  
path_img = os.listdir('Pets_Breeds/Dog/test/Afghan/01.jpg')  
pred = model.predict(path_img)  
print(pred)
```

```
pred = pred.argmax()  
  
return self.classes[pred]
```



PROCESAMIENTO DE LENGUAJE NATURAL

- **Tecnología:** DialogFlow

+ info

» Add user expression

» ¿Cómo puedo **ayudar**?

» ¿Teneis perros y gatos para adoptar?

» Háblame de tu **protectora**

 Add user expression

Frases similares

Háblame de la protectora	Háblame de la protectora, que podrías decirme de la protectora, Que me cuentas sobre vosotros, Háblame sobre vosotros
¿Teneis perros y gatos para adoptar?	Qué teneis para adoptar?, Qué hay disponible?
¿Qué necesitan las mascotas que están en tu protectora?	¿Qué necesitan las mascotas que están en tu protectora?
¿Cómo puedo ayudar?	¿Cómo puedo ayudar?, podría ayudar?, se puede ayudar en la protectora?

[Click here to edit entry](#)

puedo ser voluntario?

- **Idiomas:** Español, Ingles y Alemán

+ info

Antes de nada agradecerte que quieras dar ese paso y segundo tienes un formulario en la pagina de Voluntariado donde puedes hacer ese proceso :)

Diga algo...

Ingles

What's the cost to adopt a pet?

The adoption fee varies depending on the animal and its needs. Please visit our website or contact us for more information about our adoption fees.

Diga algo...

Alemán

"Was kann ich tun, um zu helfen?

Es gibt viele Möglichkeiten, unserem Tierheim zu helfen. Sie können eine Spende machen, Ihre Zeit als Freiwilliger zur Verfügung stellen oder sogar eines unserer Tiere adoptieren. Besuchen Sie unsere Website oder kontaktieren Sie uns für weitere Informationen.

Diga algo...



APLICACIÓN
WEB

INICIO

Bienvenidos a la protectora P.E.T!

Somos una organización malagueña sin ánimo de lucro que se encarga de cuidar gatos y perros y darles un nuevo hogar.

Adopción

Si eres un protectoras contáctanos con un proceso de adopción muy sencillo y transparente. Si estás interesado en adoptar un animal, visita nuestra sección de Adoptar un animal para conocer los requisitos y los pasos a seguir.



Gatos y perros rescatados

Colabora con nosotros

Si queréis colaborar con nuestras protectoras, tenéis varias opciones:

- Hacer donaciones para ayudarnos a cubrir los gastos de alimentación, medicación, etc.
- Convirtiros en voluntarios y ayudarnos con las tareas diarias en la protectora.
- Compartir nuestras imágenes en las redes sociales para ayudarnos a llegar a más personas.

Síguenos en redes sociales

Siguenos en nuestras redes sociales para estar al tanto de las últimas noticias y novedades de nuestras protectoras:

- [Facebook](#)
- [Twitter](#)
- [Instagram](#)

ADOPCIÓN

Gato

¿De qué raza te gustaría adoptar?

Maine Coon

Habías seleccionado adoptar un Maine Coon

Aquí hay alguna foto de los Maine Coon disponibles para su adopción:



Maine Coon Imagen

Nombre: Toby

Descripción: es un animal muy cariñoso y peludito

DONACION

Donar:

Gracias por considerar una donación a nuestro refugio de animales.

Todas las donaciones serán utilizadas para:

- Proporcionar atención médica a los animales
- Comprar alimentos y suministros
- Construir nuevas instalaciones en el refugio

Si prefieres donar en especie, estas son algunas opciones:

- Alimentos para animales
- Mantas
- Juguetes

Por favor, póngase en contacto con nosotros si desea donar alguno de estos elementos.

¿Cuánto te gustaría donar?

0

- +

[Donar](#)

Has seleccionado donar \$0

Solo puedes donar con Paypal:



Además, ¡nos encantaría contar con tu ayuda como voluntario!

Si estás interesado en ser voluntario, por favor ponte en contacto con nosotros.

CONTACTO

Contacto:

Para cualquier consulta, por favor contáctenos en el siguiente formulario:

Nombre

Email

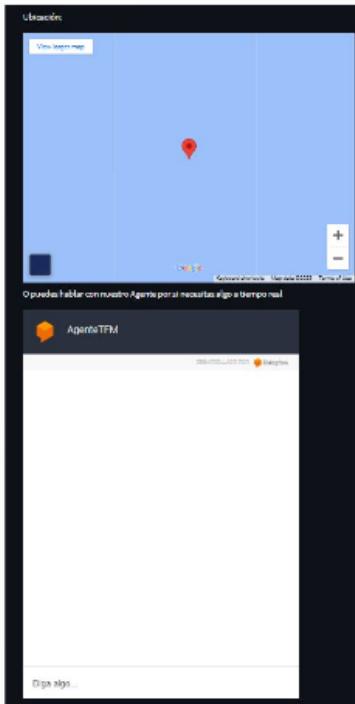
Mensaje

[Enviar mensaje](#)

O puedes contactarnos directamente:

- Teléfono: 555-1234
- Dirección: Calle Principal 123
- Horario: Lunes a Viernes de 9am a 5pm

UBICACION Y CHATBOT



VOLUNTARIADO

+ info

Voluntariado

¡Únete a nuestro equipo de voluntarios y ayúdanos a mejorar el mundo!
Rellena el siguiente formulario y nos pondremos en contacto contigo pronto.

Nombre completo

E-mail

Teléfono

Dirección

Ciudad

Pronación

País

Área de interés en tu voluntariado

Disponibilidad
 Tiempo completo

MongoDB

```
_id: ObjectId('6408c5ce18a75163af5bd038')
nombre: "Pepito el de las puertas"
email: "prueba@gmail.com"
telefono: "123456789"
direccion: "avenida de la piruleta"
ciudad: "Narnia"
provincia: "La comarca"
pais: "La ciudad de las nubes"
intereses: "No hacer nada"
disponibilidad: "Tiempo completo"
```

Gmail



pruebapet262@gmail.com <pruebapet262@gmail.com>
para mí ▾

Hola encargado del voluntariado,

Te informamos que Pepito el de las puertas se ha registrado como voluntario en nuestra organización.

Su dirección de correo electrónico es: prueba@gmail.com

¡Gracias!

ADMINISTRACIÓN

+ info

Iniciar sesión

Contraseña

Iniciar sesión

<https://pet-animal-shelter.streamlit.app/>

Drop image here
Limit: 200MB per file • JPG, JPEG, PNG

Maine_Coon_05.jpg 9.0KB



Toby

Descripción

es un animal cariñoso y peludito

Guardar

Prediction accuracy
7-day predictability



73%
accuracy

La raza es Maine Coon

Datos guardados

AUTORES



FLORES CHAMIZO SERRANO



DIEGO VALENZUELA PEREZ

-

¡GRACIAS POR SU ATENCIÓN!

¡Y RECUERDA PONERNOS BUENA NOTA!