En este tercer se hará el test del modelo tanto desde Google colaboratory como desde el ambiente virtual que se creó en el paso 1.

Para esto, siguiendo las instrucciones del documento anterior, en la misma carpeta del proyecto, cree un documento nuevo de Google colaboratory y nómbrelo **test.ipynb**.

En este archivo también tendrá que vincular Google colab con Google drive

```
from google.colab import drive
drive.mount('/content/drive')
```

y modificar el current path para la carpeta del proyecto:

```
cd '/content/drive/MyDrive/PI_MLProject'
```

Nota: estos dos comandos se deben ejecutar en celdas diferentes.

Después, en una celda nueva, defina las librerías que va a utilizar:

```
import tensorflow as tf
from tensorflow import keras
from PIL import Image
import numpy as np
import os
```

En una celda nueva, cargue el modelo previamente entrenado, tanto su arquitectura como sus pesos:

```
with open('model_config.json') as json_file:
    json_config = json_file.read()

model = keras.models.model_from_json(json_config)
model.load_weights('pets_xception_transferlearning.h5')
```

En otro celda, defina el path de los datos:

```
data_path =
'/content/drive/MyDrive/PI_MLProject/Data/cats_vs_dogs_small'
```

y en otra celda defina el path de la imagen con la que va a probar el modelo:

```
set_ = 'test'
file_ = 'dog.6946.jpg'
file_path = os.path.join(data_path,set_,file_)
```

```
print(file_path)
```

Cargue la imagen y aplique el mismo pre-procesamiento que utilizó con las imágenes de train (mismos pasos que en train):

```
image = tf.keras.preprocessing.image.load_img(file_path,target_size =
(150,150,3))
input_arr = tf.keras.preprocessing.image.img_to_array(image)
input_arr = np.array([input_arr])  # Convert single image to a batch.
```

Finalmente, entregue la imagen pre-procesada al modelo y traduzca la salida del modelo (número real) en la probabilidad de pertenencia de la imagen a la clase *"perro"*. Si esta probabilidad es mayor a 0.5, el modelo predice que la imagen es *"perro",* en caso contrario, predice que la imagen es *"gato".*

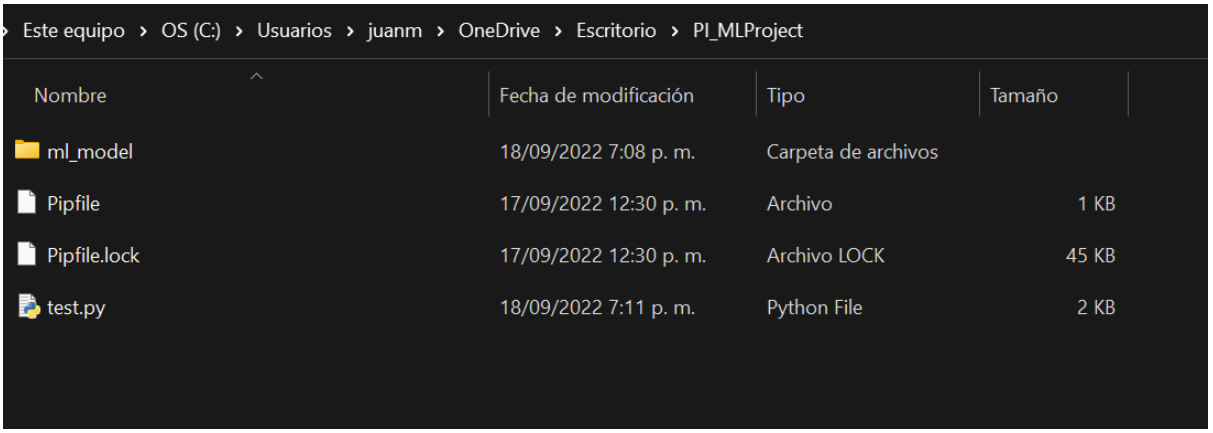
```
pred = tf.keras.activations.sigmoid(model.predict(input_arr))
if pred < 0.5:
  label = 'cat'
  prob = 1-pred
else:
  label = 'dog'
  prob = pred

print(f'The pet is a {label} with probability {prob}')
```

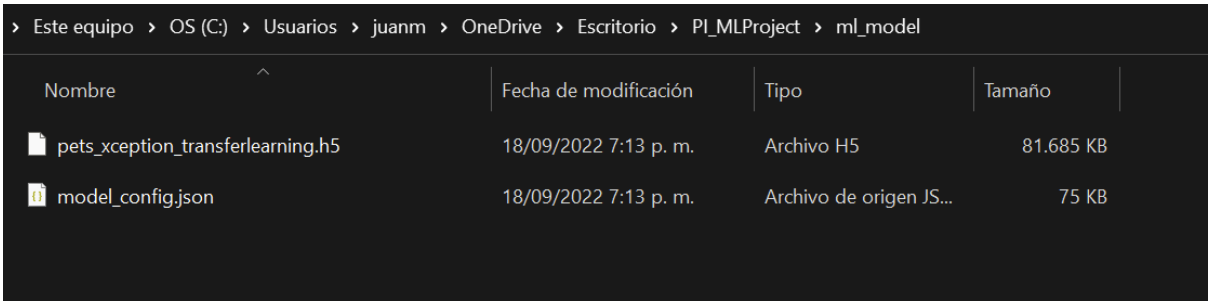
Para este ejemplo, debería ver la siguiente respuesta:

The pet is a dog with probability [[0.9796749]]

En este momento, tiene el modelo desplegado en Google colab. Para que también funcione en el ambiente virtual, descargue el archivo test.ipyn como test.py y almacénelo en la carpeta del ambiente virtual (local). Para esto vamos a Archivo -> Descargar -> Descargar.py. Nota: Es importante que estemos seguros de que no estamos descargando el jupyter notebook (.ipynb) para que el resto del proceso funcione.

Adicionalmente, tendrá que descargar los archivos model_config.json y pets_xception_transferlearning.h5 y guardarlos en la carpeta del ambiente virtual. Para esto, cree una subcarpeta con nombre ml_model





En un editor de texto (puede ser el Visual Studio Code) abra el archivo tetst.py y haga las siguientes modificaciones:

- Borre las lineas de vinculación entre google drive y google colab y de modificación del current path.
- Importe la librería sys.
- Haga el cambio sugerido con la librería os.

En esta parte, el archivo debería verse de la siguiente forma:

```
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'

import tensorflow as tf
from tensorflow import keras
from PIL import Image
import numpy as np
import sys
```

La segunda línea desactiva unos warnings generados por tensorflow dado que se está ejecutando un proceso sin GPU.

La librería sys permite recibir desde python parámetros enviados desde la consola.

Por tanto, los paths locales de los archivos del modelo y de la imagen que vamos a procesar los enviará de esta forma:

```
json_config_path = sys.argv[1]
weights_path = sys.argv[2]
file_path = sys.argv[3]

with open(json_config_path) as json_file:
    json_config = json_file.read()

model = keras.models.model_from_json(json_config)
model.load_weights(weights_path)
```

También, tendrá que borrar las lineas de código que organizaban el path de la imagen de test dado que entra como parámetro.

El archivo completo debe quedar de la siguiente forma:

```
# -*- coding: utf-8 -*-
"""test.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1foV9N_zYsLJOBwTwPWmgEHck-07ilAzD
"""

import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'

import tensorflow as tf
from tensorflow import keras
from PIL import Image
import numpy as np
import sys
```

```
json_config_path = sys.argv[1]
weights_path = sys.argv[2]
file_path = sys.argv[3]

with open(json_config_path) as json_file:
    json_config = json_file.read()

model = keras.models.model_from_json(json_config)
model.load_weights(weights_path)

image = tf.keras.preprocessing.image.load_img(file_path,target_size =
(150,150,3))
input_arr = tf.keras.preprocessing.image.img_to_array(image)
input_arr = np.array([input_arr])  # Convert single image to a batch.

pred = tf.keras.activations.sigmoid(model.predict(input_arr))
if pred < 0.5:
  label = 'cat'
  prob = 1-pred
else:
  label = 'dog'
  prob = pred

print(f'The pet is a {label} with probability {prob}')
```

Ahora, desde la terminal diríjase hasta la carpeta del ambiente virtual

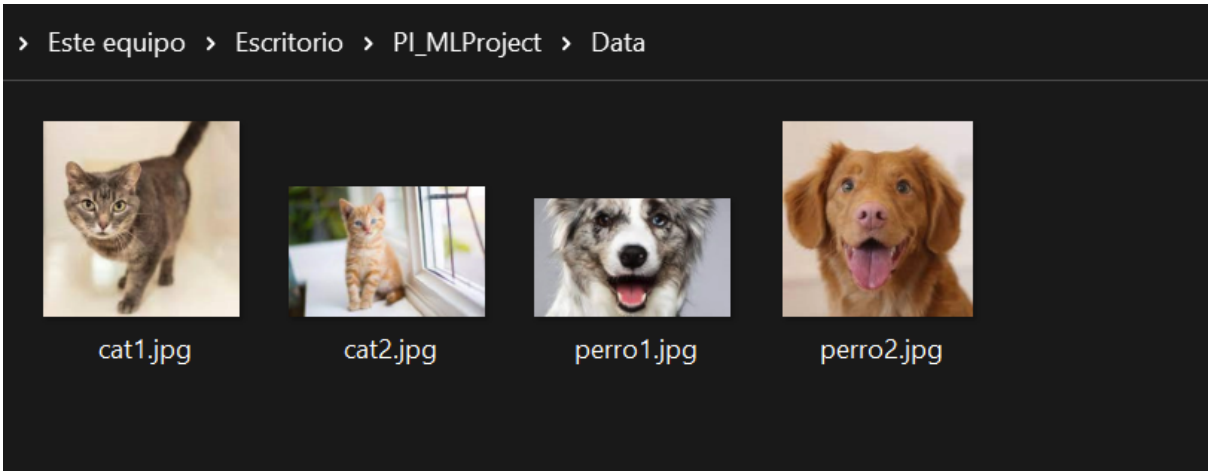
`cd C:\Users\juanm\OneDrive\Escritorio\PI_MLProject`

Active el ambiente virtual:

`pipenv shell`

```
Símbolo del sistema - pipenv  shell
Microsoft Windows [Versión 10.0.22000.856]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\juanm>cd C:\Users\juanm\OneDrive\Escritorio\PI_MLProject

C:\Users\juanm\OneDrive\Escritorio\PI_MLProject>pipenv shell
Launching subshell in virtual environment...
Microsoft Windows [Versión 10.0.22000.856]
(c) Microsoft Corporation. Todos los derechos reservados.

(PI_MLProject-xSoj08WY) C:\Users\juanm\OneDrive\Escritorio\PI_MLProject>
```

En la carpeta del ambiente virtual, cree una subcarpeta **data** donde pondrá algunas fotos de perros y gatos:



Ahora en la consola de comandos, escriba lo siguiente:

```
python test.py ml_model/model_config.json
ml_model/pets_xception_transferlearning.h5 Data/cat1.jpg
```

```
Símbolo del sistema - pipenv  shell

(PI_MLProject-xSoj08WY) C:\Users\juanm\OneDrive\Escritorio\PI_MLProject>python test.py ml_model/model_config.json ml_mod
el/pets_xception_transferlearning.h5 Data/cat1.jpg
1/1 [==============================] - 1s 668ms/step
The pet is a cat with probability [[0.99945164]]

(PI_MLProject-xSoj08WY) C:\Users\juanm\OneDrive\Escritorio\PI_MLProject>
```

Haga varios intentos modificando la imagen de prueba