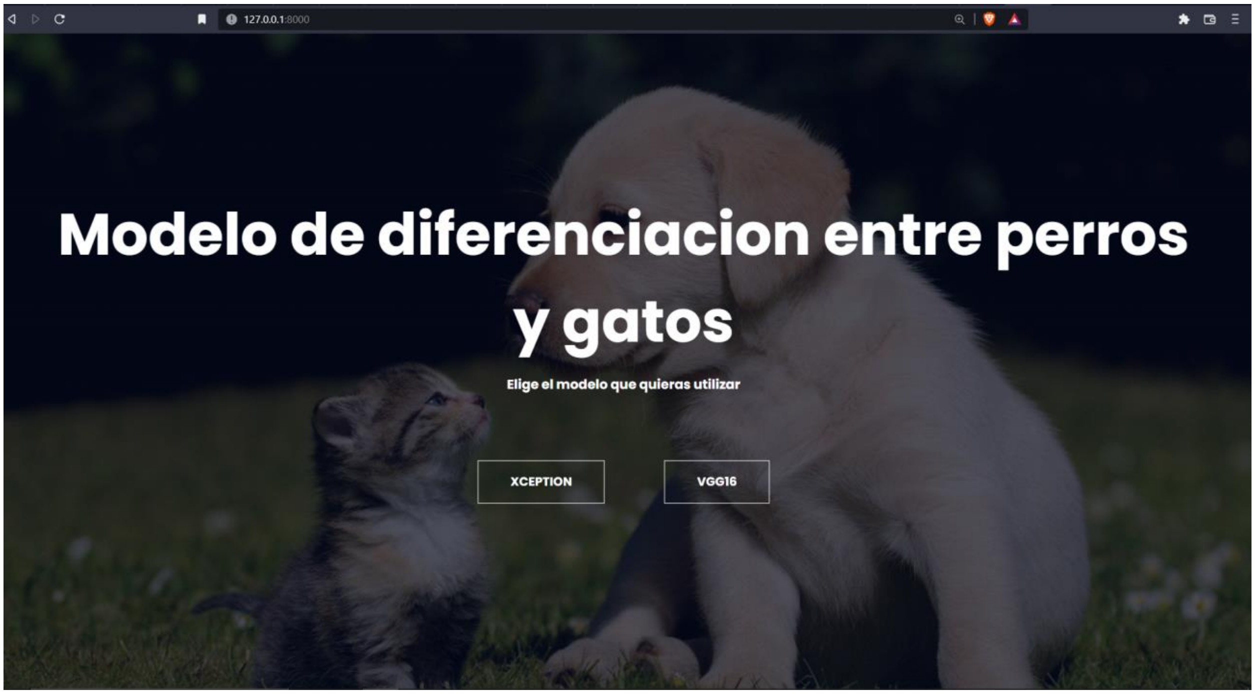
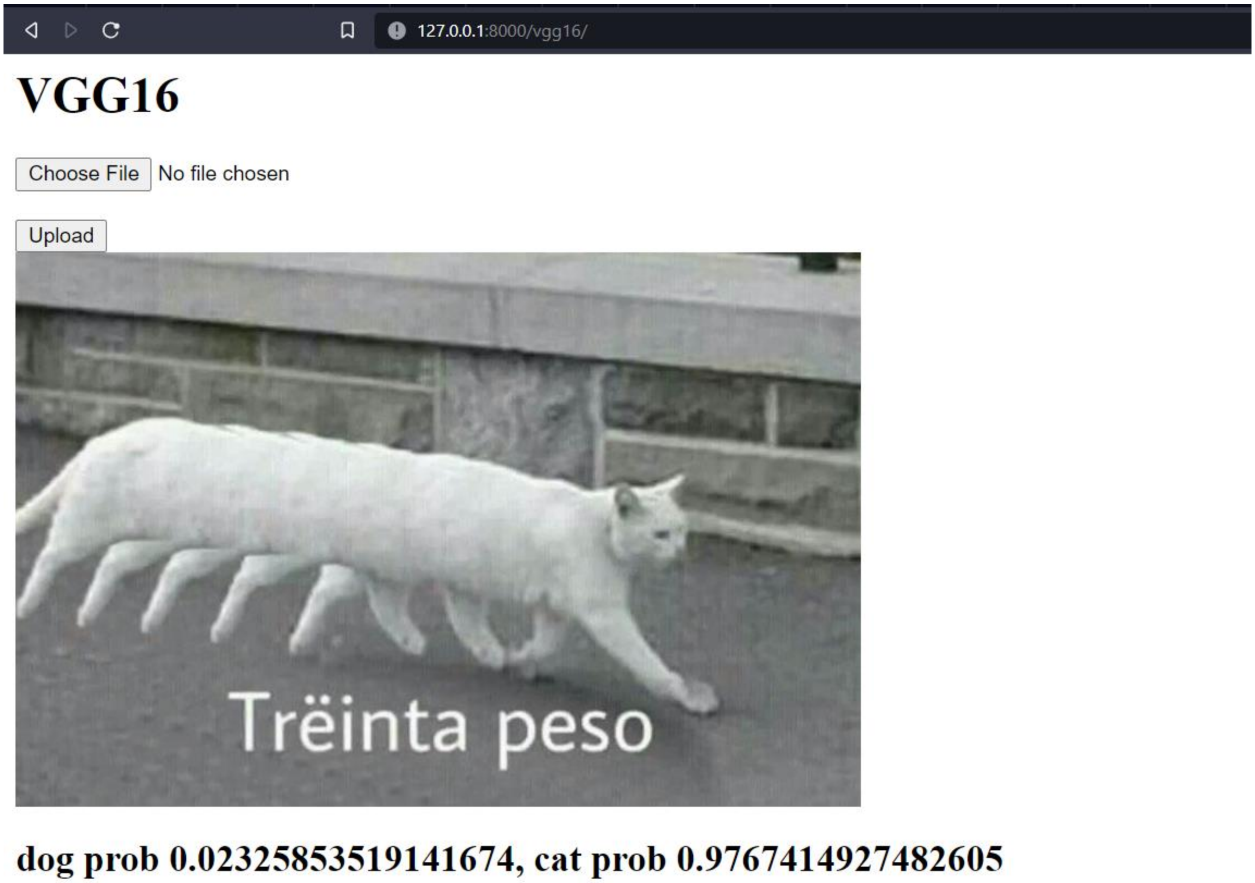


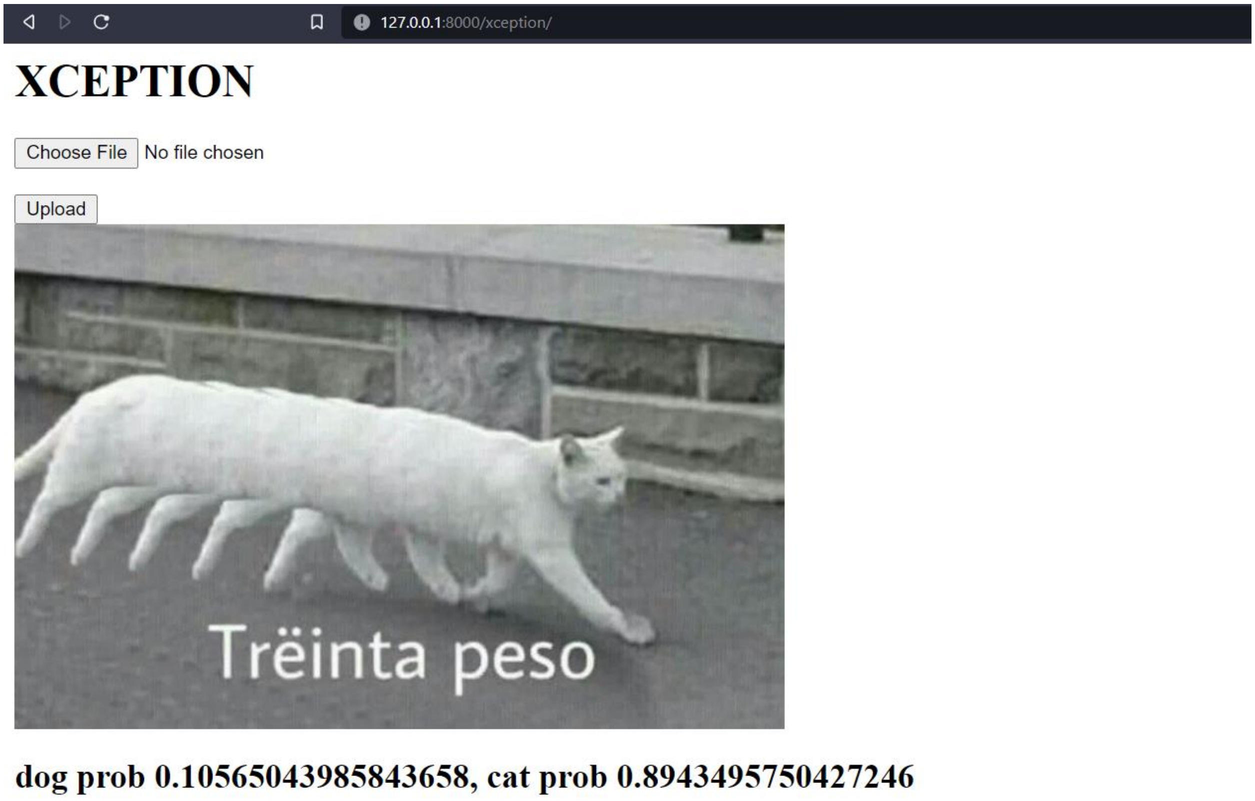
Modelo de diferenciacion entre perros y gatos

Elige el modelo que quieras utilizar



VGG16

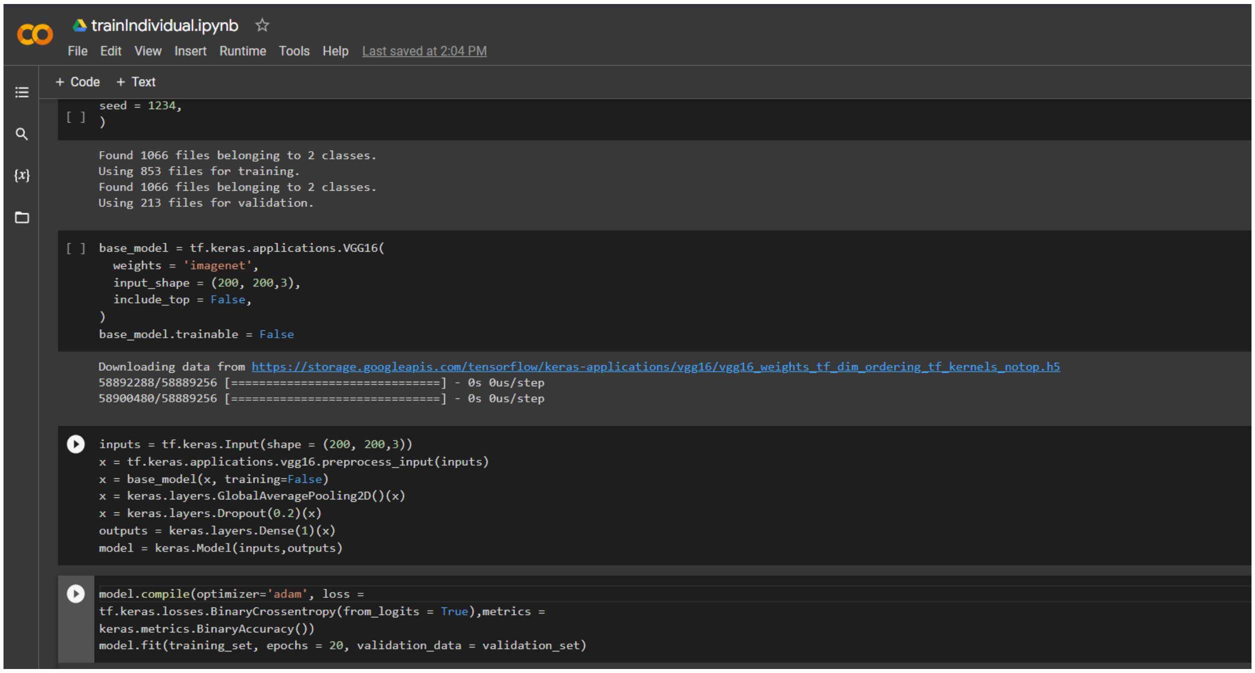
```
dog prob 0.02325853519141674, cat prob 0.9767414927482605
```



XCEPTION

dog prob 0.10565043985843658, cat prob 0.8943495750427246

MODELO ESCOGIDO POR MI EN FORMATO .ipynb



```
seed = 1234,
)
```

```
Found 1066 files belonging to 2 classes.
Using 853 files for training.
Found 1066 files belonging to 2 classes.
Using 213 files for validation.
```

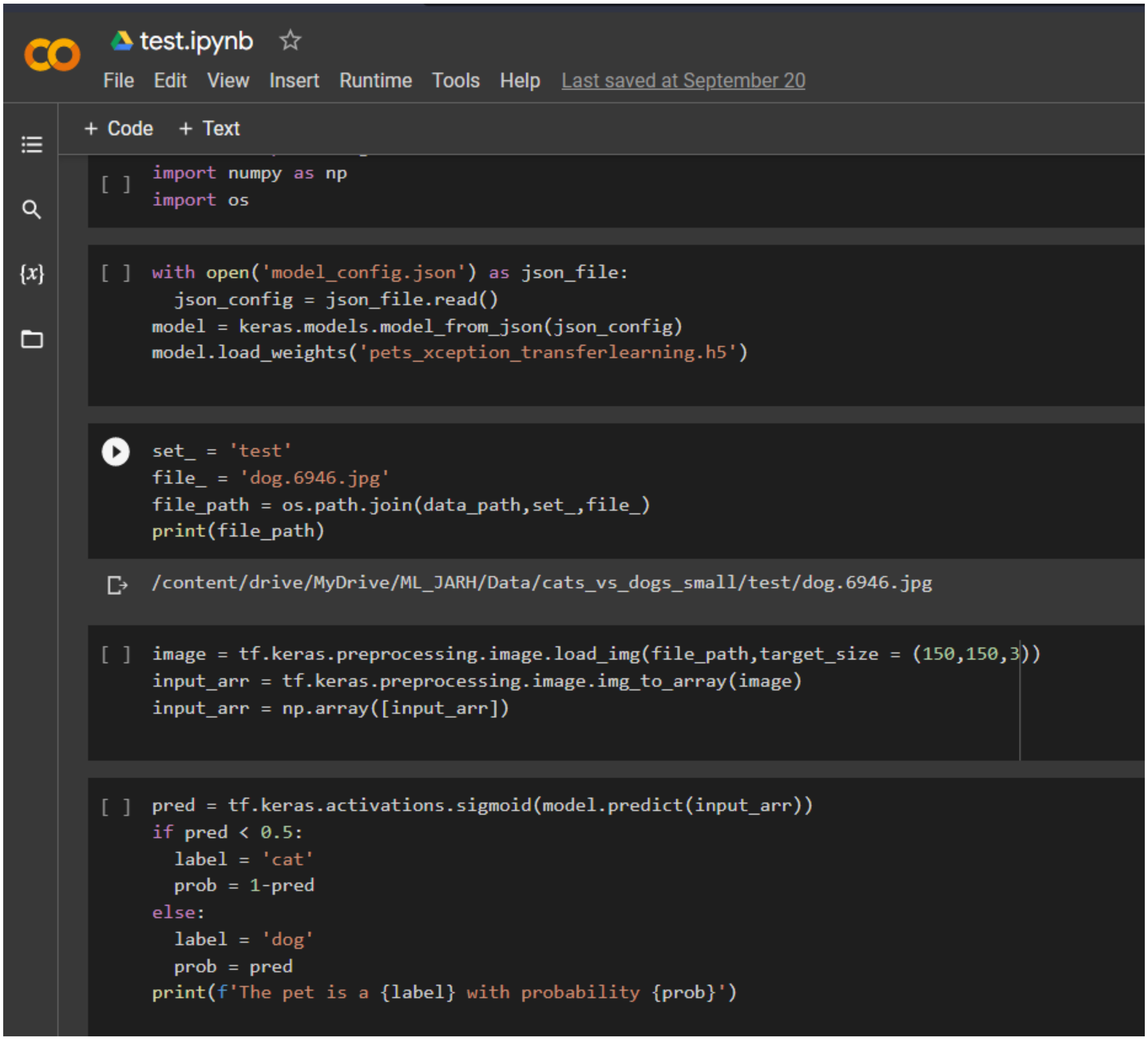
```
base_model = tf.keras.applications.VGG16(
  weights = 'imagenet',
  input_shape = (200, 200,3),
  include_top = False,
)
base_model.trainable = False
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58892288/58889256 [==============================] - 0s 0us/step
58900480/58889256 [==============================] - 0s 0us/step
```

```
inputs = tf.keras.Input(shape = (200, 200,3))
x = tf.keras.applications.vgg16.preprocess_input(inputs)
x = base_model(x, training=False)
x = keras.layers.GlobalAveragePooling2D()(x)
x = keras.layers.Dropout(0.2)(x)
outputs = keras.layers.Dense(1)(x)
model = keras.Model(inputs,outputs)
```

```
model.compile(optimizer='adam', loss =
tf.keras.losses.BinaryCrossentropy(from_logits = True),metrics =
keras.metrics.BinaryAccuracy())
model.fit(training_set, epochs = 20, validation_data = validation_set)
```

MODELO .ipynb


test.ipynb
File Edit View Insert Runtime Tools Help
Last saved at September 20
+ Code + Text

```
import numpy as np
import os
```

```
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')
```

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

```
/content/drive/MyDrive/ML_JARH/Data/cats_vs_dogs_small/test/dog.6946.jpg
```

```
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])
```

```
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}')
```