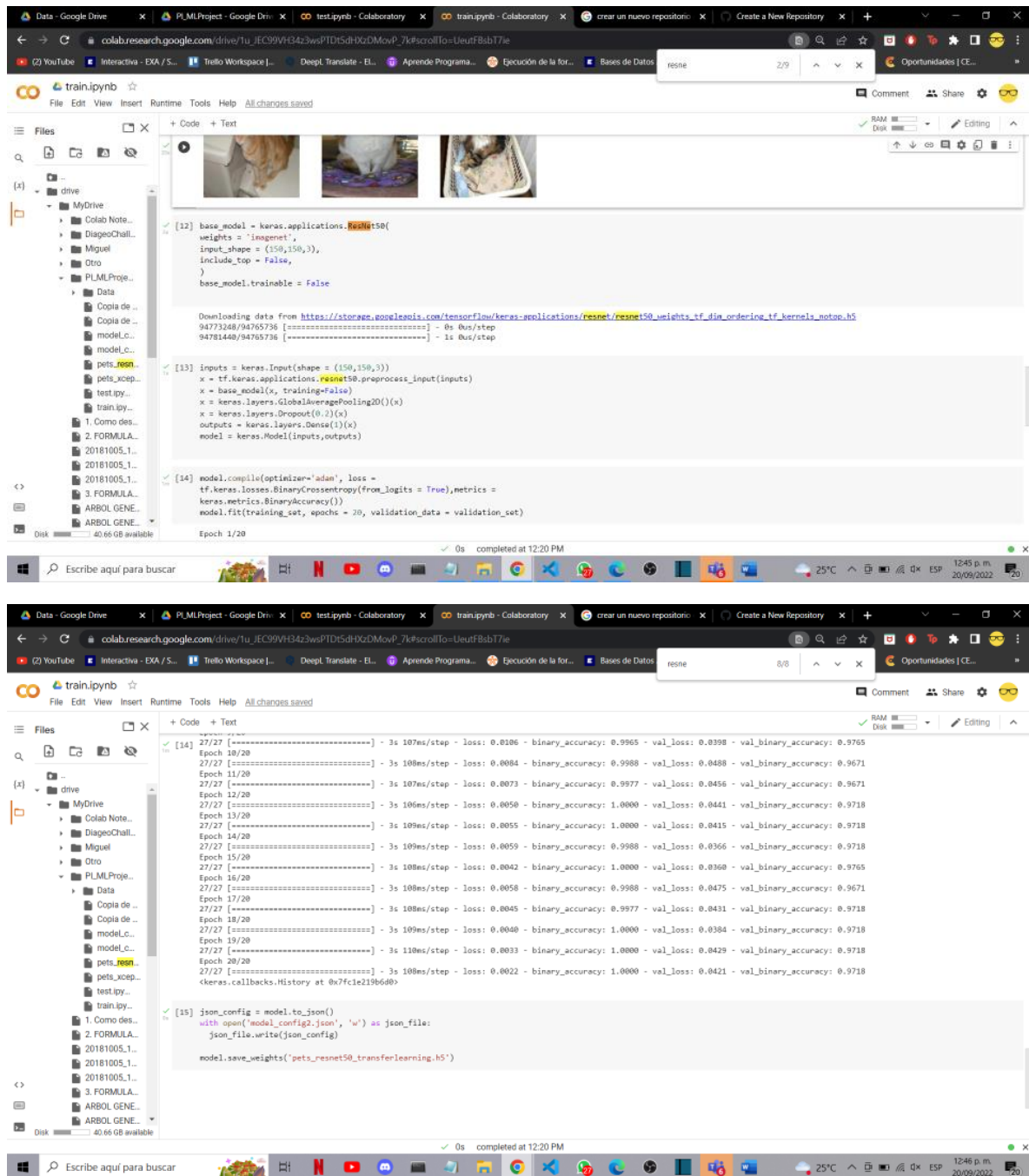


2. código train.ipynb



The screenshot displays a Google Colab notebook titled 'train.ipynb'. The notebook is open in a web browser, showing the file explorer on the left and the code editor in the center. The code is written in Python and uses Keras and TensorFlow to train a ResNet50 model. The notebook includes the following code cells:

```
[12] base_model = keras.applications.ResNet50(
      weights = 'imagenet',
      input_shape = (150, 150, 3),
      include_top = False,
      )
      base_model.trainable = False

      Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5
      94773248/94765736 [=====] - 0s 0us/step
      94781440/94765736 [=====] - 1s 0us/step

[13] inputs = keras.Input(shape = (150, 150, 3))
      x = tf.keras.applications.ResNet50.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)

[14] 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)

Epoch 1/20
```

The output of the training process is shown in the code editor, displaying the progress of the training over 20 epochs. The output includes the following information:

```
[14] 27/27 [=====] - 3s 107ms/step - loss: 0.0106 - binary_accuracy: 0.9965 - val_loss: 0.0398 - val_binary_accuracy: 0.9765
      Epoch 10/20
      27/27 [=====] - 3s 108ms/step - loss: 0.0084 - binary_accuracy: 0.9988 - val_loss: 0.0488 - val_binary_accuracy: 0.9671
      Epoch 11/20
      27/27 [=====] - 3s 107ms/step - loss: 0.0073 - binary_accuracy: 0.9977 - val_loss: 0.0456 - val_binary_accuracy: 0.9671
      Epoch 12/20
      27/27 [=====] - 3s 106ms/step - loss: 0.0050 - binary_accuracy: 1.0000 - val_loss: 0.0441 - val_binary_accuracy: 0.9718
      Epoch 13/20
      27/27 [=====] - 3s 109ms/step - loss: 0.0055 - binary_accuracy: 1.0000 - val_loss: 0.0415 - val_binary_accuracy: 0.9718
      Epoch 14/20
      27/27 [=====] - 3s 108ms/step - loss: 0.0059 - binary_accuracy: 0.9988 - val_loss: 0.0366 - val_binary_accuracy: 0.9718
      Epoch 15/20
      27/27 [=====] - 3s 108ms/step - loss: 0.0042 - binary_accuracy: 1.0000 - val_loss: 0.0368 - val_binary_accuracy: 0.9765
      Epoch 16/20
      27/27 [=====] - 3s 108ms/step - loss: 0.0058 - binary_accuracy: 0.9988 - val_loss: 0.0475 - val_binary_accuracy: 0.9671
      Epoch 17/20
      27/27 [=====] - 3s 108ms/step - loss: 0.0045 - binary_accuracy: 0.9977 - val_loss: 0.0431 - val_binary_accuracy: 0.9718
      Epoch 18/20
      27/27 [=====] - 3s 109ms/step - loss: 0.0040 - binary_accuracy: 1.0000 - val_loss: 0.0384 - val_binary_accuracy: 0.9718
      Epoch 19/20
      27/27 [=====] - 3s 110ms/step - loss: 0.0033 - binary_accuracy: 1.0000 - val_loss: 0.0429 - val_binary_accuracy: 0.9718
      Epoch 20/20
      27/27 [=====] - 3s 108ms/step - loss: 0.0022 - binary_accuracy: 1.0000 - val_loss: 0.0421 - val_binary_accuracy: 0.9718
      <keras.callbacks.History at 0x7fc1e219b6d0>

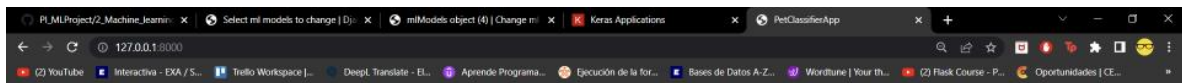
[15] json_config = model.to_json()
      with open('model_config2.json', 'w') as json_file:
          json_file.write(json_config)

      model.save_weights('pets_resnet50_transferlearning.h5')
```

The notebook also shows the file explorer on the left, which includes the following files:

- Colab Note...
- DiagesChall...
- Miguel
- Otro
- PLMLProje...
- Data
- Copia de ...
- Copia de ...
- model_c...
- model_c...
- pets_resn...
- pets_xcep...
- test.py...
- train.py...
- 1. Como des...
- 2. FORMULA...
- 20181005.1...
- 20181005.1...
- 3. FORMULA...
- ARBOL GENE...
- ARBOL GENE...

3. App Django usando ResNet50



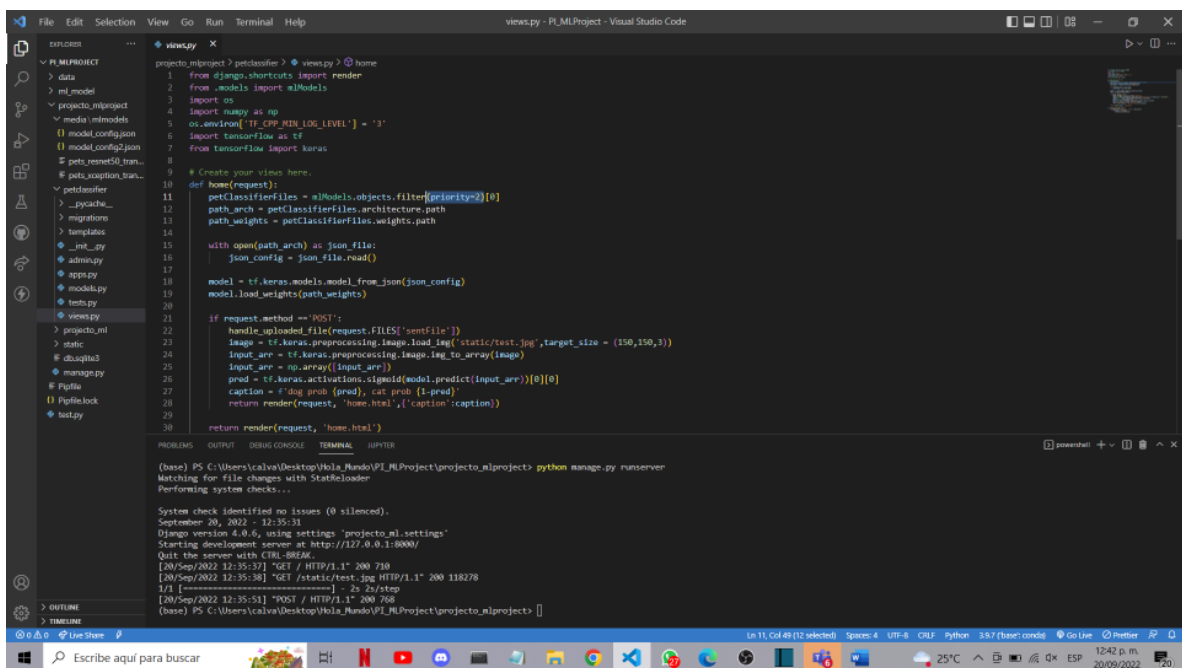
Welcome to the Pet Classifier App

Seleccionar activo: ninguno activo selec.

Upload



dog prob 0.008565174415707588, cat prob 0.9914348125457764



The screenshot shows the Django administration interface for the 'petclassifier' app. The page is titled 'Change ml models' and displays the 'mlModels object (4)'. The page layout includes a sidebar with navigation links for 'Authentication and Authorization', 'Groups', 'Users', 'PetClassifier', and 'ML models'. The main content area contains a form with the following fields:

- Title:** ML ResNet50
- Description:** ML model to classify pets and cats with ResN
- Architecture:** Currently: mmodels/model_config2.json. Change: Seleccionar archivo. Ninguno a..hivo selec.
- Weights:** Currently: mmodels/pets_resnet50_transferlearning.h5. Change: Seleccionar archivo. Ninguno a..hivo selec.
- Priority:** 2

At the bottom of the form, there are three buttons: 'Delete', 'Save and add another', and 'Save and continue editing', followed by a 'SAVE' button.