```
Jupyter Brain Tumor Last Checkpoint: 29 minutes ago (autosaved)
                                                                                                                           Logout
                                                                                                                Python 3 (ipykernel) O
                                               Widgets
                                                        Help
                                                                                                       Trusted
                   Insert Cell
                                        Langs
                                Kernel
                                                    V Autosave interval (min): off

→ □ □ ✓ ✓
                          ► Run ■ C → Code
     In [53]: import cv2
              from tensorflow import keras
             from tensorflow.keras.preprocessing import image
              from tensorflow.keras.models import Model, Sequential
              from tensorflow.keras.layers import Input, Dense, Flatten, Dropout, BatchNormalization
              from tensorflow.keras.layers import Conv2D, SeparableConv2D, MaxPool2D, LeakyReLU, Activation
              from tensorflow.keras.optimizers import Adam
              from tensorflow.keras.preprocessing.image import ImageDataGenerator
              from tensorflow.keras.callbacks import ModelCheckpoint, ReduceLROnPlateau, EarlyStopping
              from keras.utils import plot_model
              import tensorflow as tf
              import pandas as pd
             import numpy as np
              import seaborn as sns
              from matplotlib import pyplot as plt
              from sklearn.preprocessing import StandardScaler
              from sklearn.model_selection import train_test_split
              from sklearn.tree import DecisionTreeClassifier
              from sklearn metrics import accuracy_score
              from sklearn.svm import SVC
              from sklearn.linear_model import LogisticRegression
              from sklearn import neighbors
              from sklearn.ensemble import RandomForestClassifier
              from sklearn.model_selection import cross_val_score
              from sklearn.metrics import f1_score
              from sklearn metrics import confusion_matrix
              from sklearn tree import DecisionTreeClassifier
              from sklearn.preprocessing import LabelEncoder
              import os
              import joblib
              from dataprep.eda import *
              from tensorflow.keras.optimizers import Adam
              from tensorflow.keras.optimizers.legacy import Adam
              executed in 16ms, finished 14:17:46 2023-03-27
     In [34]: train = ImageDataGenerator(rescale = 1./255,
                                              shear_range = 0.2,
                                              zoom_range = 0.2,
                                              horizontal_flip = True)
              training_set = train.flow_from_directory('C:/Users/Mua/Downloads/data/brain mri/brain_tumor_train/',
                                                           target_size = (64, 64),
                                                           batch_size = 2,
                                                           class_mode = 'binary')
              executed in 51ms, finished 14:03:06 2023-03-27
              Found 253 images belonging to 2 classes.
     In [37]: test = ImageDataGenerator(rescale = 1./255,
                                              shear_range = 0.2,
                                              zoom_range = 0.2,
                                              horizontal_flip = True)
              testing_set = test.flow_from_directory('C:/Users/Mua/Downloads/data/brain mri/brain_tumor test',
                                                           target_size = (64, 64),
                                                           batch_size = 2,
                                                           class_mode = 'binary')
              executed in 55ms, finished 14:04:54 2023-03-27
              Found 253 images belonging to 2 classes.
     In [38]: training_set.class_indices
              executed in 23ms, finished 14:04:58 2023-03-27
     Out[38]: {'no': 0, 'yes': 1}
     In [39]: model = keras.models.Sequential([
                 keras.layers.Conv2D(filters=64, kernel_size=(3,3), activation='relu', input_shape=(64,64,3)),
                 keras.layers.MaxPooling2D((2, 2)),
                 keras.layers.Conv2D(filters=64,kernel_size=(3,3),activation='relu'),
                 keras.layers.MaxPooling2D((2, 2)),
                 keras.layers.Flatten(input_shape=(64,64)),
                 keras.layers.Dense(128,activation='relu'),
                 keras.layers.Dense(64,activation='relu'),
                 keras.layers.Dense(32,activation='relu'),
                 keras.layers.Dropout(rate = 0.2),
                 keras.layers.Dense(2,activation='softmax')
              executed in 185ms, finished 14:05:02 2023-03-27
     In [40]: model.compile(optimizer='adam',
                           loss='sparse_categorical_crossentropy',
                          metrics=['accuracy'])
              executed in 20ms, finished 14:05:05 2023-03-27
     In [41]: model.fit(training_set,validation_data=testing_set, epochs=10)
              executed in 2m 12s, finished 14:07:19 2023-03-27
              Epoch 1/10
              acy: 0.6719
              Epoch 2/10
              acy: 0.6877
              Epoch 3/10
              acy: 0.7194
              Epoch 4/10
              acy: 0.7470
              Epoch 5/10
              cy: 0.7075
              Epoch 6/10
              cy: 0.6877
              Epoch 7/10
              cy: 0.6957
              Epoch 8/10
              cy: 0.7549
              Epoch 9/10
              cy: 0.8142
              Epoch 10/10
              acy: 0.7747
     Out[41]: <keras.callbacks.History at 0x204e8206fa0>
     In [42]: joblib.dump(model, "Brain Tumor MRI Dataset detection.joblob")
              executed in 438ms, finished 14:08:02 2023-03-27
              Keras weights file (<HDF5 file "variables.h5" (mode r+)>) saving:
              ...layers\conv2d
              ....vars
              . . . . . . . . . . 0
              . . . . . . . . . . 1
              ...layers\conv2d_1
              ....vars
              . . . . . . . . . 0
              . . . . . . . . . . 1
              ...layers\dense
              ....vars
              . . . . . . . . . . 0
              . . . . . . . . . . 1
              ...layers\dense_1
              ....vars
              . . . . . . . . . . 0
              . . . . . . . . . . 1
              ...layers\dense_2
              ....vars
              . . . . . . . . . . 0
              . . . . . . . . . . 1
              ...layers\dense_3
              ....vars
              . . . . . . . . . . 0
              .....1
              ...layers\dropout
              ....vars
              ...layers\flatten
              ....vars
              ...layers\max_pooling2d
              ....vars
              ...layers\max_pooling2d_1
              ....vars
              ...metrics\mean
              ....vars
              . . . . . . . . . 0
              ...metrics\mean_metric_wrapper
              ....vars
              . . . . . . . . . . 0
              . . . . . . . . . . 1
              ...optimizer
              ....vars
              . . . . . . . . . 0
              . . . . . . . . . . 10
              .....13
              . . . . . . . . . 14
              . . . . . . . . . . 15
              . . . . . . . . . . 16
              . . . . . . . . . . 17
              . . . . . . . . . . . 18
              . . . . . . . . . . 19
              . . . . . . . . . 2
              . . . . . . . . . 20
              .....21
              . . . . . . . . . . 22
              . . . . . . . . . 3
              . . . . . . . . . 4
              . . . . . . . . . 5
              . . . . . . . . . 6
              . . . . . . . . . 8
              . . . . . . . . . . 9
              ...vars
              Keras model archive saving:
                                                               Modified
                                                                                  Size
              File Name
                                                                                  4130
              config.json
                                                         2023-03-27 14:08:01
              metadata.json
                                                         2023-03-27 14:08:01
                                                                                     64
              variables.h5
                                                         2023-03-27 14:08:01
                                                                               19901624
     Out[42]: ['Brain Tumor MRI Dataset detection.joblob']
     In [43]: model = joblib.load("Brain Tumor MRI Dataset detection.joblob")
              executed in 547ms, finished 14:08:08 2023-03-27
              Keras model archive loading:
                                                               Modified
                                                                                  Size
              File Name
              config.json
                                                         2023-03-27 14:08:00
                                                                                   4130
              metadata.json
                                                                                     64
                                                         2023-03-27 14:08:00
              variables.h5
                                                                               19901624
                                                         2023-03-27 14:08:00
              Keras weights file (<HDF5 file "variables.h5" (mode r)>) loading:
              ...layers\conv2d
              ....vars
              . . . . . . . . . 0
              . . . . . . . . . . 1
              ...layers\conv2d_1
              ....vars
              . . . . . . . . . 0
              . . . . . . . . . . . 1
              ...layers\dense
              ....vars
              . . . . . . . . . 0
              ......1
              ...layers\dense_1
              ....vars
              . . . . . . . . . . 0
              . . . . . . . . . . 1
              ...layers\dense_2
              ....vars
              .......0
              .....1
              ...layers\dense_3
              ....vars
              . . . . . . . . . 0
              . . . . . . . . . . 1
              ...layers\dropout
              ....vars
              ...layers\flatten
              ....vars
              ...layers\max_pooling2d
              ....vars
              ...layers\max_pooling2d_1
              ....vars
              ...metrics\mean
              ....vars
              . . . . . . . . . 0
              ...metrics\mean_metric_wrapper
              ....vars
              . . . . . . . . . 0
              . . . . . . . . . . . 1
              ...optimizer
              ....vars
              . . . . . . . . . . . . . . . . . 12
              . . . . . . . . . . 15
              .....16
              . . . . . . . . . . 17
              . . . . . . . . . . . 18
              .....20
              . . . . . . . . . 21
              .....22
              . . . . . . . . . 23
              . . . . . . . . . 4
              . . . . . . . . . . 5
              . . . . . . . . . . 7
              ...vars
     In [44]: def getlabel(x):
                 for i in training_set.class_indices:
                     if training_set.class_indices[i] == x:
                         return i
              executed in 11ms, finished 14:08:11 2023-03-27
     In [45]: def predict_image(path):
                  img = image.load_img(path,target_size=(64,64))
                 img = np.expand_dims(img, axis=0)
                 p = model.predict(img)
                 pred = [np.argmax(element) for element in p]
                 print (plt.imshow(cv2.imread(path)))
                 return getlabel(pred[0])
              executed in 14ms, finished 14:08:14 2023-03-27
     In [47]: predict_image("C:/Users/Mua/Downloads/mri_brain.webp")
              executed in 509ms, finished 14:14:14 2023-03-27
              1/1 [======= ] - 0s 44ms/step
              AxesImage(80,52.8;496x369.6)
     Out[47]: 'yes'
               100 -
               200 -
               300 -
               400
               500
               600
               700 -
                      100 200 300 400 500 600
     In [46]: predict_image("C:/Users/Mua/Downloads/ri.webp")
              executed in 546ms, finished 14:14:09 2023-03-27
              1/1 [======= ] - 0s 135ms/step
              AxesImage(80,52.8;496x369.6)
     Out[46]: 'yes'
               100 -
               200 -
               300 -
               400 -
               500 -
                               200
                                             400
                                      300
                                                    500
     In [48]: predict_image("C:/Users/Mua/Downloads/mri.png")
              executed in 390ms, finished 14:14:44 2023-03-27
              1/1 [======] - 0s 48ms/step
              AxesImage(80,52.8;496x369.6)
     Out[48]: 'yes'
                50 -
               100 -
               250 -
                                100
                                       150
                                              200
                                                      250
     In [50]: predict_image("C:/Users/Mua/Downloads/mrii.jpg")
              executed in 484ms, finished 14:16:17 2023-03-27
              1/1 [======] - 0s 39ms/step
              AxesImage(80,52.8;496x369.6)
     Out[50]: 'no'
               100
               120
               140
               160
                                                                   250
                                               150
                                                         200
     In [51]: model.summary()
              executed in 52ms, finished 14:16:58 2023-03-27
              Model: "sequential_2"
                                        Output Shape
               Layer (type)
                                                                Param #
              _____
               conv2d_4 (Conv2D)
                                         (None, 62, 62, 64)
                                                                1792
               max_pooling2d_4 (MaxPooling (None, 31, 31, 64)
                                                                0
               2D)
               conv2d_5 (Conv2D)
                                         (None, 29, 29, 64)
                                                                36928
               max_pooling2d_5 (MaxPooling (None, 14, 14, 64)
                                                                0
               2D)
               flatten_2 (Flatten)
                                         (None, 12544)
                                                                0
               dense_8 (Dense)
                                        (None, 128)
                                                                1605760
                                                                8256
               dense_9 (Dense)
                                        (None, 64)
               dense_10 (Dense)
                                        (None, 32)
                                                                2080
               dropout_2 (Dropout)
                                        (None, 32)
                                                                0
               dense_11 (Dense)
                                         (None, 2)
                                                                66
              ______
              Total params: 1,654,882
              Trainable params: 1,654,882
              Non-trainable params: 0
      In [ ]:
```