```
# IMPORTANT: RUN THIS CELL IN ORDER TO IMPORT YOUR KAGGLE DATA SOURCES
# TO THE CORRECT LOCATION (/kaggle/input) IN YOUR NOTEBOOK,
# THEN FEEL FREE TO DELETE THIS CELL.
# NOTE: THIS NOTEBOOK ENVIRONMENT DIFFERS FROM KAGGLE'S PYTHON
# ENVIRONMENT SO THERE MAY BE MISSING LIBRARIES USED BY YOUR
# NOTEBOOK.

import os
import sys
from tempfile import NamedTemporaryFile
from urllib.request import urlopen
from urllib.parse import unquote, urlparse
from urllib.error import HTTPError
from zipfile import ZipFile
import tarfile
import shutil

CHUNK_SIZE = 40960
DATA_SOURCE_MAPPING = 'brain-mri-images-for-brain-tumor-detection:https%3A%2F%2Fstorage.googleapis.com%2Fkaggle-data-sets%2F165566%2F3771

KAGGLE_INPUT_PATH='/kaggle/input'
KAGGLE_WORKING_PATH='/kaggle/working'
KAGGLE_SYMLINK='kaggle'

!umount /kaggle/input/ 2> /dev/null
shutil.rmtree('/kaggle/input', ignore_errors=True)
os.makedirs(KAGGLE_INPUT_PATH, 0o777, exist_ok=True)
os.makedirs(KAGGLE_WORKING_PATH, 0o777, exist_ok=True)

try:
  os.symlink(KAGGLE_INPUT_PATH, os.path.join("..", 'input'), target_is_directory=True)
except FileExistsError:
  pass
try:
  os.symlink(KAGGLE_WORKING_PATH, os.path.join("..", 'working'), target_is_directory=True)
except FileExistsError:
  pass

for data_source_mapping in DATA_SOURCE_MAPPING.split(','):
    directory, download_url_encoded = data_source_mapping.split(':')
    download_url = unquote(download_url_encoded)
    filename = urlparse(download_url).path
    destination_path = os.path.join(KAGGLE_INPUT_PATH, directory)
    try:
        with urlopen(download_url) as fileres, NamedTemporaryFile() as tfile:
            total_length = fileres.headers['content-length']
            print(f'Downloading {directory}, {total_length} bytes compressed')
            dl = 0
            data = fileres.read(CHUNK_SIZE)
            while len(data) > 0:
                dl += len(data)
                tfile.write(data)
                done = int(50 * dl / int(total_length))
                sys.stdout.write(f"\r[{'=' * done}{' ' * (50-done)}] {dl} bytes downloaded")
                sys.stdout.flush()
                data = fileres.read(CHUNK_SIZE)
            if filename.endswith('.zip'):
              with ZipFile(tfile) as zfile:
                zfile.extractall(destination_path)
            else:
              with tarfile.open(tfile.name) as tarfile:
                tarfile.extractall(destination_path)
            print(f'\nDownloaded and uncompressed: {directory}')
    except HTTPError as e:
        print(f'Failed to load (likely expired) {download_url} to path {destination_path}')
        continue
    except OSError as e:
        print(f'Failed to load {download_url} to path {destination_path}')
        continue

print('Data source import complete.')
```

```
Downloading brain-mri-images-for-brain-tumor-detection, 15828590 bytes compressed
[==================================================] 15828590 bytes downloaded
Downloaded and uncompressed: brain-mri-images-for-brain-tumor-detection
Data source import complete.
```

Brain Tumor Detection

CNN models:

- VGG19
- Inceptionv3

1. Loads libraries

```
%matplotlib inline
from IPython import display
import os
import math
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import tensorflow as tf
from sklearn.metrics import *
from tensorflow import keras
from keras.preprocessing.image import ImageDataGenerator

np.random.seed(42)
tf.random.set_seed(42)
```

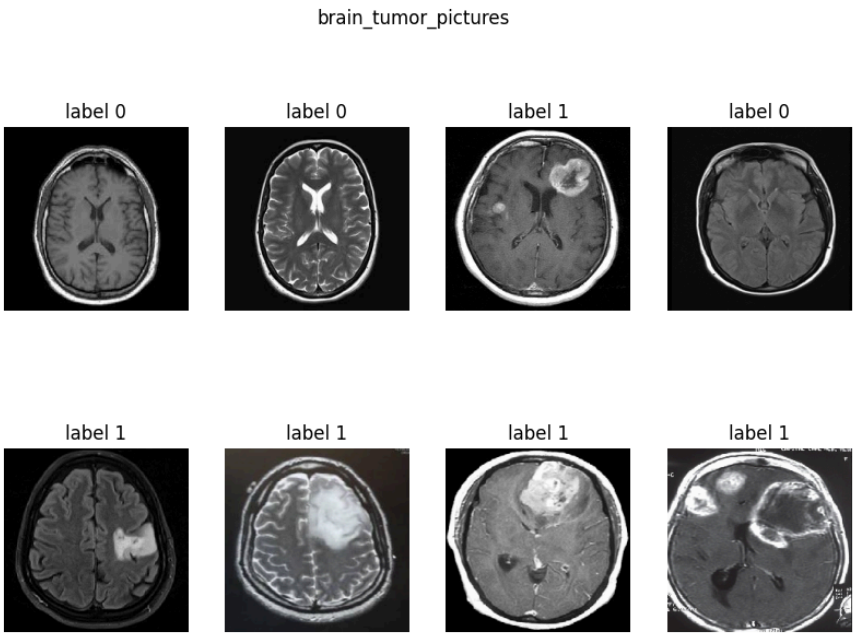
2. Loads Images

```
#Setting some inital parameters
height, width = 224, 224
batch_size=64


data_dir = '/kaggle/input/brain-mri-images-for-brain-tumor-detection/'

def image_generator(height,width):
    datagen = ImageDataGenerator(
            rescale=1./255.,
            validation_split=0.2,
            )
    train_ds = datagen.flow_from_directory(
            data_dir,
            batch_size=batch_size,
            subset="training",
            #color_mode = 'grayscale',
            shuffle=True,
            class_mode='binary',
            target_size=(height, width),
            classes={'no': 0., 'yes': 1.}
            )
    val_ds = datagen.flow_from_directory(
              data_dir,
              subset="validation",
              #seed=123,
              #color_mode = 'grayscale',
              class_mode='binary',
              target_size=(height, width),
              batch_size=batch_size,
            classes={'no': 0., 'yes': 1.}
            )
    return train_ds, val_ds

train_ds, val_ds = image_generator(height,width)

total_image = np.concatenate([train_ds.labels,val_ds.labels])
print('\n\n',{'No_brain_tumor_cases':len(np.where(total_image==0)[0]),
      'brain_tumor_cases':len(np.where(total_image==1)[0])})
```

```
Found 203 images belonging to 2 classes.
Found 50 images belonging to 2 classes.


 {'No_brain_tumor_cases': 98, 'brain_tumor_cases': 155}
```

3. Image demonstration

```
fig, ax = plt.subplots(2, 4, figsize=(10, 7))
fig.suptitle("brain_tumor_pictures")
for k in range(8):
    images, labels = train_ds.next()
    i, j = k//4, k%4
    ax[i, j].imshow(images[0])
    ax[i, j].set_title(f'label {int(labels[0])}')
    ax[i, j].axis('off')
plt.show()
```



4. CNN Implementation

```
from tensorflow.keras.layers.experimental import preprocessing

tf.keras.backend.clear_session()
input_shape = (height, width, 3)
base_model = tf.keras.applications.vgg19.VGG19(
    weights='imagenet',
    include_top=False,
    input_shape=input_shape
)
base_model.trainable = False

model_vgg19 = tf.keras.Sequential()
model_vgg19.add(base_model)
model_vgg19.add(tf.keras.layers.Flatten())

model_vgg19.add(tf.keras.layers.Dense(1, activation='sigmoid'))

model_vgg19.compile(loss='binary_crossentropy',
              optimizer=tf.keras.optimizers.Adam(0.01),
              metrics=['acc'])
model_vgg19.summary()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg19/vgg19_weights_tf_dim_ordering_tf_kernels_no
80134624/80134624 [==============================] - 0s 0us/step
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg19 (Functional)          (None, 7, 7, 512)         20024384

 flatten (Flatten)           (None, 25088)             0

 dense (Dense)               (None, 1)                 25089

=================================================================
```

```
Total params: 20049473 (76.48 MB)
Trainable params: 25089 (98.00 KB)
Non-trainable params: 20024384 (76.39 MB)
_________________________________________________________________
```

```
checkpoint = tf.keras.callbacks.ModelCheckpoint('model/vgg19_best.h5', monitor='acc', verbose=1, mode='max',save_best_only=True)
early = tf.keras.callbacks.EarlyStopping(monitor="acc", mode="max",restore_best_weights=True, patience=5)
callbacks_list = [checkpoint,early]

history = model_vgg19.fit(
        train_ds,
        validation_data=val_ds,
        epochs=15,
        shuffle=True,
        verbose=True,
        callbacks=callbacks_list)
```

```
4/4 [==============================] - 181s 58s/step - loss: 0.8161 - acc: 0.8325 - val_loss: 1.9710 - val_acc: 0.8000
Epoch 7/25
4/4 [==============================] - ETA: 0s - loss: 0.3985 - acc: 0.8966
Epoch 7: acc improved from 0.85222 to 0.89655, saving model to model/vgg19_best.h5
4/4 [==============================] - 169s 54s/step - loss: 0.3985 - acc: 0.8966 - val_loss: 1.0941 - val_acc: 0.7800
Epoch 8/25
4/4 [==============================] - ETA: 0s - loss: 0.3648 - acc: 0.9064
Epoch 8: acc improved from 0.89655 to 0.90640, saving model to model/vgg19_best.h5
4/4 [==============================] - 175s 56s/step - loss: 0.3648 - acc: 0.9064 - val_loss: 1.1900 - val_acc: 0.8000
Epoch 9/25
4/4 [==============================] - ETA: 0s - loss: 0.4105 - acc: 0.8867
Epoch 9: acc did not improve from 0.90640
4/4 [==============================] - 181s 46s/step - loss: 0.4105 - acc: 0.8867 - val_loss: 0.6491 - val_acc: 0.8600
Epoch 10/25
4/4 [==============================] - ETA: 0s - loss: 0.1878 - acc: 0.9261
Epoch 10: acc improved from 0.90640 to 0.92611, saving model to model/vgg19_best.h5
4/4 [==============================] - 167s 42s/step - loss: 0.1878 - acc: 0.9261 - val_loss: 0.4749 - val_acc: 0.8800
Epoch 11/25
4/4 [==============================] - ETA: 0s - loss: 0.2142 - acc: 0.9360
Epoch 11: acc improved from 0.92611 to 0.93596, saving model to model/vgg19_best.h5
4/4 [==============================] - 176s 44s/step - loss: 0.2142 - acc: 0.9360 - val_loss: 0.6377 - val_acc: 0.8600
Epoch 12/25
4/4 [==============================] - ETA: 0s - loss: 0.1323 - acc: 0.9458
Epoch 12: acc improved from 0.93596 to 0.94581, saving model to model/vgg19_best.h5
4/4 [==============================] - 175s 56s/step - loss: 0.1323 - acc: 0.9458 - val_loss: 0.7112 - val_acc: 0.8200
Epoch 13/25
4/4 [==============================] - ETA: 0s - loss: 0.0354 - acc: 0.9901
Epoch 13: acc improved from 0.94581 to 0.99015, saving model to model/vgg19_best.h5
4/4 [==============================] - 178s 57s/step - loss: 0.0354 - acc: 0.9901 - val_loss: 1.0724 - val_acc: 0.8200
Epoch 14/25
4/4 [==============================] - ETA: 0s - loss: 0.1223 - acc: 0.9507
Epoch 14: acc did not improve from 0.99015
4/4 [==============================] - 168s 42s/step - loss: 0.1223 - acc: 0.9507 - val_loss: 0.5433 - val_acc: 0.8800
Epoch 15/25
4/4 [==============================] - ETA: 0s - loss: 0.0051 - acc: 1.0000
Epoch 15: acc improved from 0.99015 to 1.00000, saving model to model/vgg19_best.h5
4/4 [==============================] - 175s 44s/step - loss: 0.0051 - acc: 1.0000 - val_loss: 0.4511 - val_acc: 0.9000
Epoch 16/25
4/4 [==============================] - ETA: 0s - loss: 0.0308 - acc: 0.9901
Epoch 16: acc did not improve from 1.00000
4/4 [==============================] - 168s 42s/step - loss: 0.0308 - acc: 0.9901 - val_loss: 0.4269 - val_acc: 0.8800
Epoch 17/25
4/4 [==============================] - ETA: 0s - loss: 0.0045 - acc: 1.0000
Epoch 17: acc did not improve from 1.00000
4/4 [==============================] - 174s 44s/step - loss: 0.0045 - acc: 1.0000 - val_loss: 0.6082 - val_acc: 0.8600
Epoch 18/25
4/4 [==============================] - ETA: 0s - loss: 0.0090 - acc: 1.0000
Epoch 18: acc did not improve from 1.00000
4/4 [==============================] - 175s 44s/step - loss: 0.0090 - acc: 1.0000 - val_loss: 0.7859 - val_acc: 0.8400
Epoch 19/25
4/4 [==============================] - ETA: 0s - loss: 0.0048 - acc: 1.0000
Epoch 19: acc did not improve from 1.00000
4/4 [==============================] - 175s 55s/step - loss: 0.0048 - acc: 1.0000 - val_loss: 0.4931 - val_acc: 0.8600
Epoch 20/25
4/4 [==============================] - ETA: 0s - loss: 0.0016 - acc: 1.0000
Epoch 20: acc did not improve from 1.00000
4/4 [==============================] - 168s 42s/step - loss: 0.0016 - acc: 1.0000 - val_loss: 0.4216 - val_acc: 0.9000
```

```
train_result = model_vgg19.evaluate(train_ds)
val_result = model_vgg19.evaluate(val_ds)

no_augmented_df = pd.DataFrame(zip(train_result,val_result),columns=['Train','Val'],index=['Loss','Acc'])
no_augmented_df
```

```
4/4 [==============================] - 135s 31s/step - loss: 0.0317 - acc: 0.9852
1/1 [==============================] - 33s 33s/step - loss: 0.4511 - acc: 0.9000
```

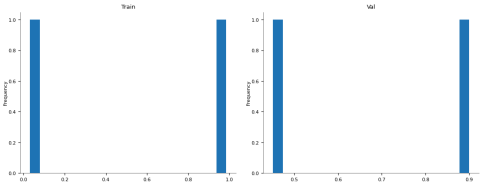
1 to 2 of 2 entries Filter

index	Train	Val
Loss	0.03172457218170166	0.4511299431324005
Acc	0.9852216839790344	0.8999999761581421

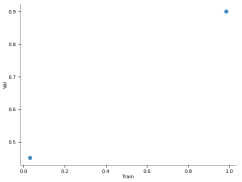
Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

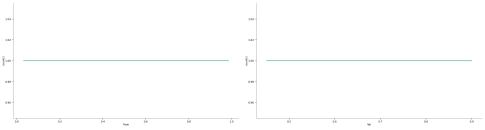
Distributions



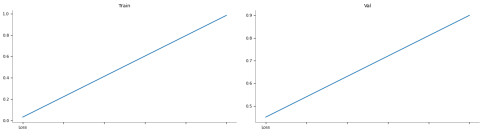
2-d distributions



Time series

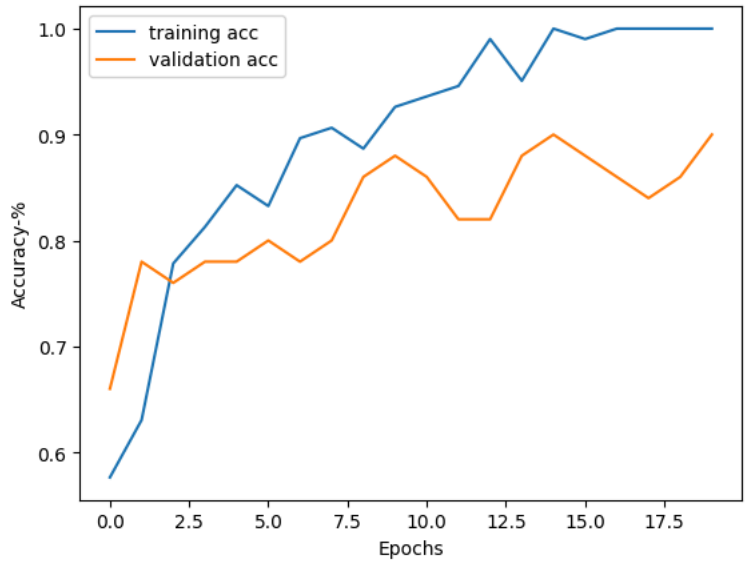


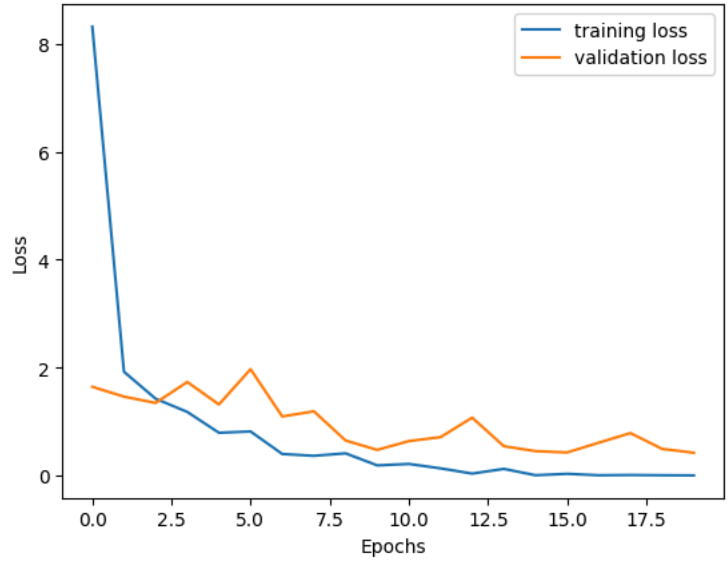
Values



Next steps: Generate code with no_augmented_df View recommended plots

```
# plot learning curve
def plot_learning_curve(history):
    acc = history.history['acc']
    val_acc = history.history['val_acc']
    loss = history.history['loss']
    val_loss = history.history['val_loss']
    epochs = range(len(acc))

    plt.plot(epochs, acc, label='training acc')
    plt.plot(epochs, val_acc, label='validation acc')
    plt.xlabel('Epochs')
    plt.ylabel('Accuracy-%')
    plt.legend()
    plt.figure()

    plt.plot(epochs, loss, label='training loss')
    plt.plot(epochs, val_loss, label='validation loss')
    plt.xlabel('Epochs')
    plt.ylabel('Loss')
    plt.legend()

plot_learning_curve(history)
```





Adding Image Augmentation

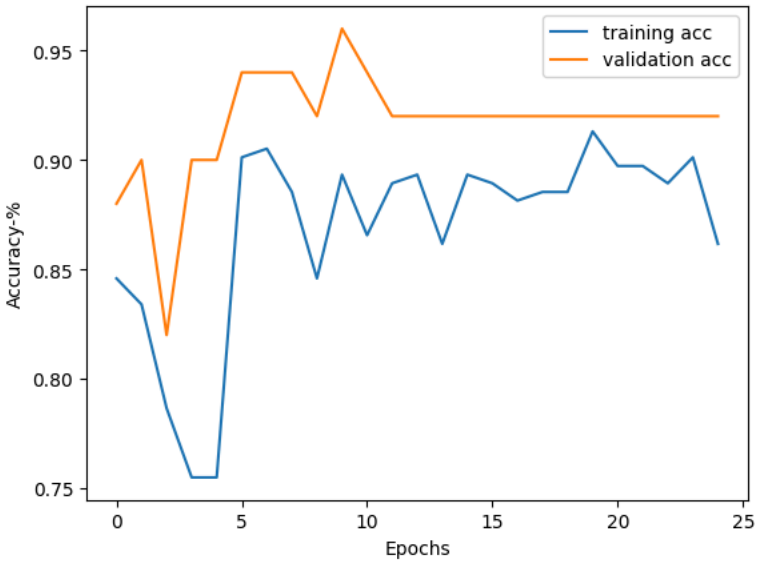
```
def augmentataion_generator(height,width):
    datagen = ImageDataGenerator(
            rescale=1./255.,
            width_shift_range=0.1,
            height_shift_range=0.1,
            shear_range=0.1,
            zoom_range=0.1,
            rotation_range=30,
            horizontal_flip=True,
            brightness_range=(0.5, 1.0)
            )
    aug_train_ds = datagen.flow_from_directory(
            data_dir,
            batch_size=64,
            shuffle=True,
            class_mode='binary',
            target_size=(height, width),
            classes={'no': 0., 'yes': 1.}
            )
    return aug_train_ds
aug_train_ds = augmentataion_generator(height,width)
```

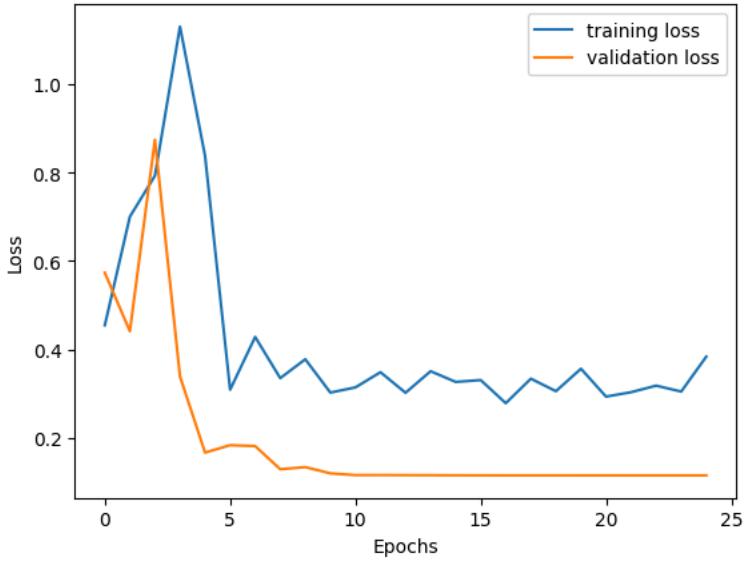
```
Found 253 images belonging to 2 classes.
```

```
reduce_lr = tf.keras.callbacks.ReduceLROnPlateau(monitor='acc', factor=0.3,
                            patience=2, min_lr=0.0000001)
callbacks_list = [checkpoint,reduce_lr]

history = model_vgg19.fit(
        aug_train_ds,
        validation_data=val_ds,
        epochs=25,
        verbose=True,
        callbacks=callbacks_list)
```

```
Epoch 11: acc did not improve from 1.00000
4/4 [==============================] - 209s 55s/step - loss: 0.3141 - acc: 0.8656 - val_loss: 0.1158 - val_acc: 0.9400 - lr: 2.7
Epoch 12/25
4/4 [==============================] - ETA: 0s - loss: 0.3485 - acc: 0.8893
Epoch 12: acc did not improve from 1.00000
4/4 [==============================] - 209s 56s/step - loss: 0.3485 - acc: 0.8893 - val_loss: 0.1159 - val_acc: 0.9200 - lr: 8.1
Epoch 13/25
4/4 [==============================] - ETA: 0s - loss: 0.3020 - acc: 0.8933
Epoch 13: acc did not improve from 1.00000
4/4 [==============================] - 210s 55s/step - loss: 0.3020 - acc: 0.8933 - val_loss: 0.1157 - val_acc: 0.9200 - lr: 8.1
Epoch 14/25
4/4 [==============================] - ETA: 0s - loss: 0.3507 - acc: 0.8617
Epoch 14: acc did not improve from 1.00000
4/4 [==============================] - 202s 52s/step - loss: 0.3507 - acc: 0.8617 - val_loss: 0.1156 - val_acc: 0.9200 - lr: 2.4
Epoch 15/25
4/4 [==============================] - ETA: 0s - loss: 0.3266 - acc: 0.8933
Epoch 15: acc did not improve from 1.00000
4/4 [==============================] - 210s 55s/step - loss: 0.3266 - acc: 0.8933 - val_loss: 0.1154 - val_acc: 0.9200 - lr: 2.4
Epoch 16/25
4/4 [==============================] - ETA: 0s - loss: 0.3307 - acc: 0.8893
Epoch 16: acc did not improve from 1.00000
4/4 [==============================] - 210s 55s/step - loss: 0.3307 - acc: 0.8893 - val_loss: 0.1153 - val_acc: 0.9200 - lr: 7.2
Epoch 17/25
4/4 [==============================] - ETA: 0s - loss: 0.2784 - acc: 0.8814
Epoch 17: acc did not improve from 1.00000
4/4 [==============================] - 208s 55s/step - loss: 0.2784 - acc: 0.8814 - val_loss: 0.1152 - val_acc: 0.9200 - lr: 7.2
Epoch 18/25
4/4 [==============================] - ETA: 0s - loss: 0.3339 - acc: 0.8854
Epoch 18: acc did not improve from 1.00000
4/4 [==============================] - 211s 56s/step - loss: 0.3339 - acc: 0.8854 - val_loss: 0.1152 - val_acc: 0.9200 - lr: 2.1
Epoch 19/25
4/4 [==============================] - ETA: 0s - loss: 0.3056 - acc: 0.8854
Epoch 19: acc did not improve from 1.00000
4/4 [==============================] - 212s 56s/step - loss: 0.3056 - acc: 0.8854 - val_loss: 0.1151 - val_acc: 0.9200 - lr: 2.1
Epoch 20/25
4/4 [==============================] - ETA: 0s - loss: 0.3563 - acc: 0.9130
Epoch 20: acc did not improve from 1.00000
4/4 [==============================] - 210s 55s/step - loss: 0.3563 - acc: 0.9130 - val_loss: 0.1151 - val_acc: 0.9200 - lr: 6.5
Epoch 21/25
4/4 [==============================] - ETA: 0s - loss: 0.2933 - acc: 0.8972
Epoch 21: acc did not improve from 1.00000
4/4 [==============================] - 210s 55s/step - loss: 0.2933 - acc: 0.8972 - val_loss: 0.1151 - val_acc: 0.9200 - lr: 6.5
Epoch 22/25
4/4 [==============================] - ETA: 0s - loss: 0.3033 - acc: 0.8972
Epoch 22: acc did not improve from 1.00000
4/4 [==============================] - 210s 55s/step - loss: 0.3033 - acc: 0.8972 - val_loss: 0.1151 - val_acc: 0.9200 - lr: 6.5
Epoch 23/25
4/4 [==============================] - ETA: 0s - loss: 0.3181 - acc: 0.8893
Epoch 23: acc did not improve from 1.00000
4/4 [==============================] - 211s 56s/step - loss: 0.3181 - acc: 0.8893 - val_loss: 0.1151 - val_acc: 0.9200 - lr: 1.9
Epoch 24/25
4/4 [==============================] - ETA: 0s - loss: 0.3047 - acc: 0.9012
Epoch 24: acc did not improve from 1.00000
4/4 [==============================] - 210s 55s/step - loss: 0.3047 - acc: 0.9012 - val_loss: 0.1151 - val_acc: 0.9200 - lr: 1.9
Epoch 25/25
4/4 [==============================] - ETA: 0s - loss: 0.3837 - acc: 0.8617
Epoch 25: acc did not improve from 1.00000
```

```
plot_learning_curve(history)
```





Final VGG19 model results

```
train_result = model_vgg19.evaluate(train_ds)
val_result = model_vgg19.evaluate(val_ds)

augmented_df = pd.DataFrame(zip(train_result,val_result),columns=['Train','Val'],index=['Loss','Acc'])
augmented_df
```

```
4/4 [==============================] - 133s 30s/step - loss: 0.0060 - acc: 1.0000
1/1 [==============================] - 34s 34s/step - loss: 0.1151 - acc: 0.9200
```

	Train	Val
Loss	0.006029	0.115098
Acc	1.000000	0.920000

Next steps: Generate code with augmented_df View recommended plots

```
ypred_val = model_vgg19.predict(val_ds[0][0])
ypred_val = np.array([1 if x > 0.5 else 0 for x in ypred_val])
y_val = val_ds[0][-1]

print(confusion_matrix(y_val, ypred_val))
print('\n',classification_report(ypred_val,y_val))
```

```
2/2 [==============================] - 34s 12s/step
[[16  3]
 [ 1 30]]

              precision    recall  f1-score   support

           0       0.84      0.94      0.89        17
```

```
                1       0.97      0.91      0.94        33

         accuracy                           0.92        50
        macro avg       0.90      0.93      0.91        50
     weighted avg       0.93      0.92      0.92        50
```

```
# inception v3
height = 299
width = 299

train_ds, val_ds = image_generator(height,width)

tf.keras.backend.clear_session()
input_shape = (height, width, 3)
base_model = tf.keras.applications.InceptionV3(
    weights='imagenet',
    include_top=False,
    input_shape=input_shape
)
base_model.trainable = False

model_inceptionv3 = tf.keras.Sequential()
model_inceptionv3.add(base_model)
model_inceptionv3.add(tf.keras.layers.Flatten())
model_inceptionv3.add(tf.keras.layers.Dense(1, activation='sigmoid'))

model_inceptionv3.compile(
                    loss='binary_crossentropy',
                    optimizer=tf.keras.optimizers.Adam(0.001),
                    metrics=['acc']
)
model_inceptionv3.summary()
```

```
Found 203 images belonging to 2 classes.
Found 50 images belonging to 2 classes.
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/inception_v3/inception_v3_weights_tf_dim_ordering
87910968/87910968 [==============================] - 0s 0us/step
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 inception_v3 (Functional)   (None, 8, 8, 2048)        21802784

 flatten (Flatten)           (None, 131072)            0

 dense (Dense)               (None, 1)                 131073

=================================================================
Total params: 21933857 (83.67 MB)
Trainable params: 131073 (512.00 KB)
Non-trainable params: 21802784 (83.17 MB)
_________________________________________________________________
```

```
# train inception v3
checkpoint = tf.keras.callbacks.ModelCheckpoint('model/inceptionv3_best.h5', monitor='acc', verbose=1, mode='max',save_best_only=True)
early = tf.keras.callbacks.EarlyStopping(monitor="acc", mode="max",restore_best_weights=True, patience=5)
callbacks_list = [checkpoint,early]

history = model_inceptionv3.fit(
        train_ds,
        validation_data=val_ds,
        epochs=25,
        verbose=True,
        callbacks=callbacks_list)
```

```
4/4 [==============================] - 64s 17s/step - loss: 2.3687 - acc: 0.7635 - val_loss: 1.5012 - val_acc: 0.8800
Epoch 4/25
4/4 [==============================] - ETA: 0s - loss: 0.6517 - acc: 0.8916
Epoch 4: acc improved from 0.76355 to 0.89163, saving model to model/inceptionv3_best.h5
```

```
4/4 [==============================] - ETA: 0s - loss: 0.0807 - acc: 0.9803
Epoch 8: acc improved from 0.95567 to 0.98030, saving model to model/inceptionv3_best.h5
4/4 [==============================] - 55s 18s/step - loss: 0.0807 - acc: 0.9803 - val_loss: 1.5795 - val_acc: 0.8800
Epoch 9/25
4/4 [==============================] - ETA: 0s - loss: 0.1458 - acc: 0.9655
Epoch 9: acc did not improve from 0.98030
4/4 [==============================] - 64s 17s/step - loss: 0.1458 - acc: 0.9655 - val_loss: 1.7459 - val_acc: 0.8800
Epoch 10/25
4/4 [==============================] - ETA: 0s - loss: 0.0596 - acc: 0.9852
Epoch 10: acc improved from 0.98030 to 0.98522, saving model to model/inceptionv3_best.h5
4/4 [==============================] - 65s 17s/step - loss: 0.0596 - acc: 0.9852 - val_loss: 1.3303 - val_acc: 0.9000
Epoch 11/25
4/4 [==============================] - ETA: 0s - loss: 0.0130 - acc: 0.9951
Epoch 11: acc improved from 0.98522 to 0.99507, saving model to model/inceptionv3_best.h5
4/4 [==============================] - 55s 14s/step - loss: 0.0130 - acc: 0.9951 - val_loss: 1.1023 - val_acc: 0.9200
Epoch 12/25
4/4 [==============================] - ETA: 0s - loss: 0.0016 - acc: 1.0000
Epoch 12: acc improved from 0.99507 to 1.00000, saving model to model/inceptionv3_best.h5
4/4 [==============================] - 66s 17s/step - loss: 0.0016 - acc: 1.0000 - val_loss: 1.0722 - val_acc: 0.9200
Epoch 13/25
4/4 [==============================] - ETA: 0s - loss: 0.0060 - acc: 0.9951
Epoch 13: acc did not improve from 1.00000
4/4 [==============================] - 65s 17s/step - loss: 0.0060 - acc: 0.9951 - val_loss: 1.0815 - val_acc: 0.9200
Epoch 14/25
4/4 [==============================] - ETA: 0s - loss: 0.0027 - acc: 1.0000
Epoch 14: acc did not improve from 1.00000
4/4 [==============================] - 65s 21s/step - loss: 0.0027 - acc: 1.0000 - val_loss: 1.0898 - val_acc: 0.9200
Epoch 15/25
4/4 [==============================] - ETA: 0s - loss: 0.0011 - acc: 1.0000
Epoch 15: acc did not improve from 1.00000
4/4 [==============================] - 64s 16s/step - loss: 0.0011 - acc: 1.0000 - val_loss: 1.0951 - val_acc: 0.9200
Epoch 16/25
4/4 [==============================] - ETA: 0s - loss: 1.8513e-04 - acc: 1.0000
Epoch 16: acc did not improve from 1.00000
4/4 [==============================] - 75s 18s/step - loss: 1.8513e-04 - acc: 1.0000 - val_loss: 1.1016 - val_acc: 0.9200
Epoch 17/25
4/4 [==============================] - ETA: 0s - loss: 1.1737e-04 - acc: 1.0000
Epoch 17: acc did not improve from 1.00000
4/4 [==============================] - 55s 14s/step - loss: 1.1737e-04 - acc: 1.0000 - val_loss: 1.1067 - val_acc: 0.9200
```

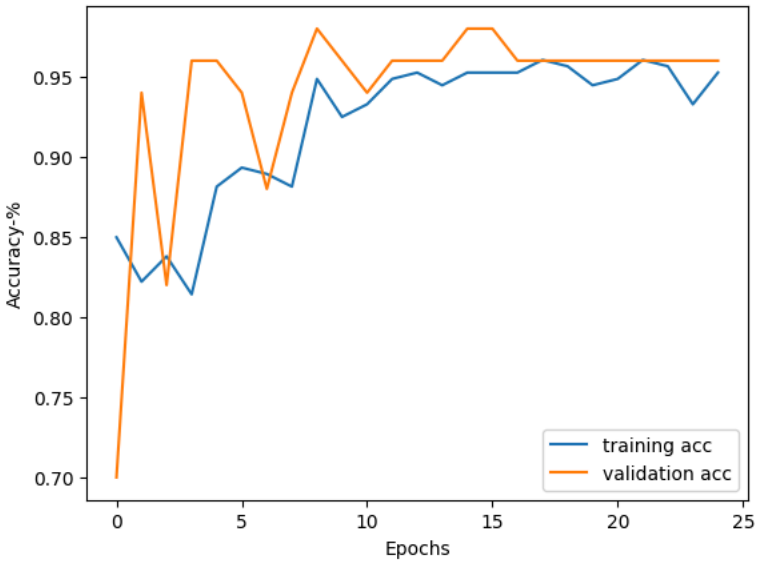
```
train_result = model_inceptionv3.evaluate(train_ds)
val_result = model_inceptionv3.evaluate(val_ds)

no_augmented_df = pd.DataFrame(zip(train_result,val_result),columns=['Train','Val'],index=['Loss','Acc'])
no_augmented_df
```

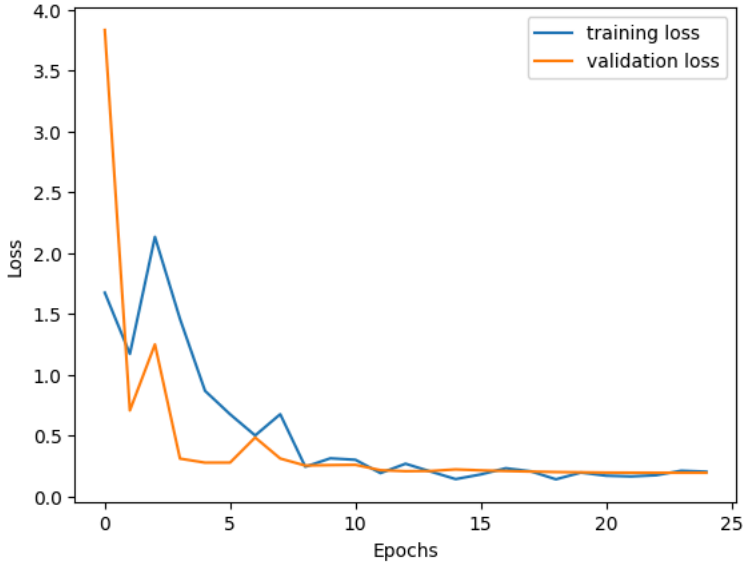
```
4/4 [==============================] - 43s 10s/step - loss: 0.0026 - acc: 1.0000
1/1 [==============================] - 11s 11s/step - loss: 1.0722 - acc: 0.9200
```

	Train	Val
Loss	0.002631	1.072156
Acc	1.000000	0.920000

Next steps: Generate code with `no_augmented_df` View recommended plots

```
plot_learning_curve(history)
```





Adding Image augmentation...

```
aug_train_ds = augmentataion_generator(height,width)

early = tf.keras.callbacks.EarlyStopping(monitor="val_loss", mode="min",restore_best_weights=True, patience=5)

reduce_lr = tf.keras.callbacks.ReduceLROnPlateau(monitor='acc', factor=0.3,
                              patience=3, min_lr=0.0000001)
callbacks_list = [checkpoint,early,reduce_lr]

history = model_inceptionv3.fit(
        aug_train_ds,
        validation_data=val_ds,
        epochs=25,
        #shuffle=True,
        verbose=True,
        callbacks=callbacks_list)
```

```
Epoch 15: acc did not improve from 1.00000
4/4 [==============================] - 73s 18s/step - loss: 0.1425 - acc: 0.9526 - val_loss: 0.2222 - val_acc: 0.9800 - lr: 9.00
Epoch 16/25
4/4 [==============================] - ETA: 0s - loss: 0.1809 - acc: 0.9526
Epoch 16: acc did not improve from 1.00000
4/4 [==============================] - 83s 21s/step - loss: 0.1809 - acc: 0.9526 - val_loss: 0.2144 - val_acc: 0.9800 - lr: 9.00
Epoch 17/25
4/4 [==============================] - ETA: 0s - loss: 0.2317 - acc: 0.9526
Epoch 17: acc did not improve from 1.00000
4/4 [==============================] - 81s 21s/step - loss: 0.2317 - acc: 0.9526 - val_loss: 0.2090 - val_acc: 0.9600 - lr: 2.70
Epoch 18/25
4/4 [==============================] - ETA: 0s - loss: 0.2065 - acc: 0.9605
Epoch 18: acc did not improve from 1.00000
4/4 [==============================] - 81s 21s/step - loss: 0.2065 - acc: 0.9605 - val_loss: 0.2045 - val_acc: 0.9600 - lr: 2.70
Epoch 19/25
4/4 [==============================] - ETA: 0s - loss: 0.1409 - acc: 0.9565
Epoch 19: acc did not improve from 1.00000
4/4 [==============================] - 81s 21s/step - loss: 0.1409 - acc: 0.9565 - val_loss: 0.2005 - val_acc: 0.9600 - lr: 2.70
Epoch 20/25
4/4 [==============================] - ETA: 0s - loss: 0.1962 - acc: 0.9447
Epoch 20: acc did not improve from 1.00000
4/4 [==============================] - 71s 18s/step - loss: 0.1962 - acc: 0.9447 - val_loss: 0.1984 - val_acc: 0.9600 - lr: 2.70
Epoch 21/25
4/4 [==============================] - ETA: 0s - loss: 0.1705 - acc: 0.9486
Epoch 21: acc did not improve from 1.00000
4/4 [==============================] - 72s 18s/step - loss: 0.1705 - acc: 0.9486 - val_loss: 0.1960 - val_acc: 0.9600 - lr: 2.70
Epoch 22/25
4/4 [==============================] - ETA: 0s - loss: 0.1646 - acc: 0.9605
Epoch 22: acc did not improve from 1.00000
4/4 [==============================] - 72s 18s/step - loss: 0.1646 - acc: 0.9605 - val_loss: 0.1953 - val_acc: 0.9600 - lr: 8.10
Epoch 23/25
4/4 [==============================] - ETA: 0s - loss: 0.1742 - acc: 0.9565
Epoch 23: acc did not improve from 1.00000
4/4 [==============================] - 81s 21s/step - loss: 0.1742 - acc: 0.9565 - val_loss: 0.1949 - val_acc: 0.9600 - lr: 8.10
Epoch 24/25
4/4 [==============================] - ETA: 0s - loss: 0.2111 - acc: 0.9328
Epoch 24: acc did not improve from 1.00000
4/4 [==============================] - 80s 21s/step - loss: 0.2111 - acc: 0.9328 - val_loss: 0.1949 - val_acc: 0.9600 - lr: 8.10
Epoch 25/25
4/4 [==============================] - ETA: 0s - loss: 0.2034 - acc: 0.9526
```

Final InceptionV3 model results

```
plot_learning_curve(history)
```