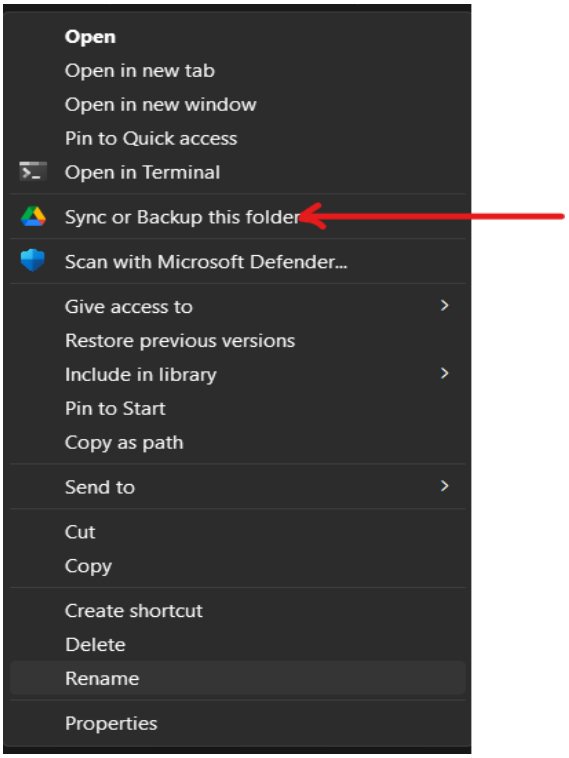
Step By Step Guide for “Insights into Celebrity Recognition: A Comparative Study”

Setting Your Google Colab:

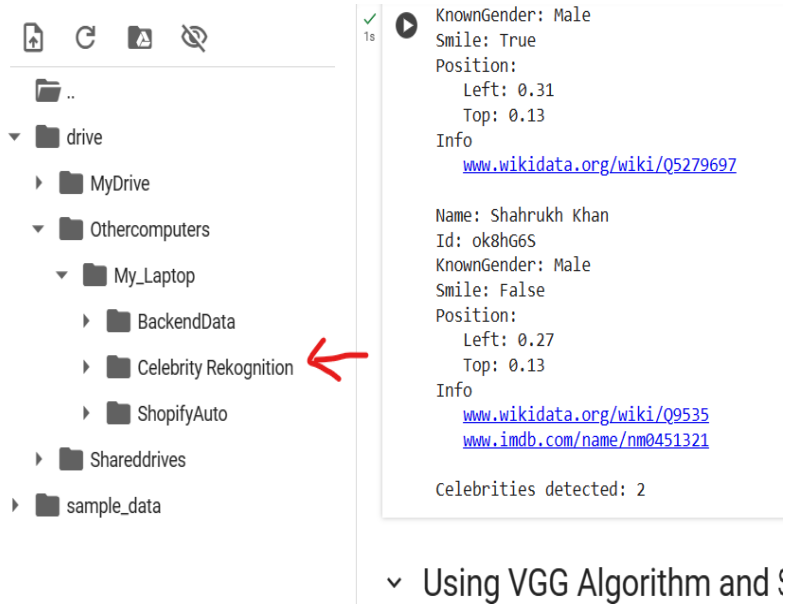
- **Install Google Drive:**
  - To get started with the project, firstly we need to take the provided dataset on google drive.
  - You can refer to this link for installing Google Drive on local machine.
- **Share Folder to Google Drive:**
  - Navigate to the D: Drive
  - Paste the “Celebrity Rekognition” Folder to D: Drive
  - Right Click on the Folder “Celebrity Rekognition” and Click on Sync and Backup.
  - If any Additional information required refer to this Link.
  - Note: Please Make sure Dataset would be fully uploaded to google drive to run the program.



- **Mount Google Drive in Notebook:**
  - In your notebook, mount Google Drive to make the drive folder visible on the left side pane using the first code snippet.
  - Once you have successfully mounted your Google Drive in Google Collab, you can confirm the successful mount by checking the left-hand pane of your Collab window. This indicates that your notebook has access to the files and folders within your Google Drive.

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

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

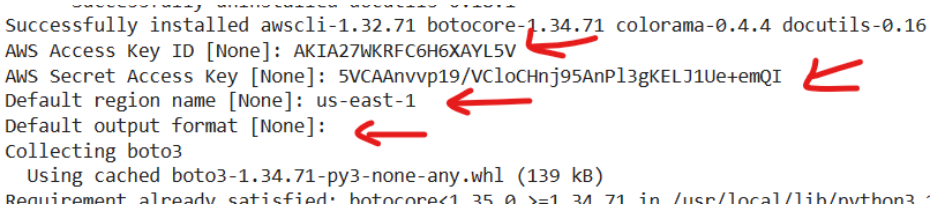


1) Steps to Run First Section of Our Project "Using AWS Reckognition: A cloud deployed service":

- After the Drive has been mounted, Run the code snippet one by one and where the first code snippet commands to set up dependencies and libraries to run the project effectively, **sometimes it can crash the Google Collab due to heavy dependencies and may require running again.**
- It asks for the inputs given in the code snippet and provided below for reliable access.

Note: Please do not share these inputs as these are user id and password to enter AWS cloud using sdks.

aws_access_key_id = 'AKIA27WKRFC6H6XAYL5V'
aws_secret_access_key = '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'
region ='us-east-1'
Output Format = Simply press enter



- Run the snippets and please provide the path of celebrity's image to be recognized into variable name "photo" from the "Dataset For AWS Recognition" Folder provided in the dataset or download any image from internet and place it into "Dataset For AWS Recognition" and try recognizing it by providing the path.

```
[13] import cv2
     from google.colab.patches import cv2_imshow
     photo = '/content/drive/Othercomputers/My_Laptop/Celebrity Rekognition/Dataset For AWS Recognition/SalmanandSRK.jpg'
     img = cv2.imread(photo)
     cv2_imshow(img)
     celeb_count = recognize_celebrities(photo)
     print("Celebrities detected: " + str(celeb_count))
```

2) Steps to Run Second Section of Our Project "Using VGG Algorithm and SVM Classifier":

- ❖ Run the code snippets one by one to have insights into the dataset and learn about the project.
- ❖ Make sure that the path to the dataset is correct to run the project effectively.

```
[19] import os
     source_dir=os.path.join('/content/drive/Othercomputers/My_Laptop/Celebrity Rekognition/MixedData')
```

```
model = vgg_face()

model.load_weights('/content/drive/Othercomputers/My_Laptop/Celebrity Rekognition/vgg-face-weights/vgg_face_weights.h5')
```

- ❖ Note: The following step will take a lot of time as it is extracting the features from the Dataset of 27000+ images.

```
embeddings = np.zeros((metadata.shape[0], 2622))
for i, m in enumerate(metadata):
    img_path = metadata[i].image_path()
    img = load_image(img_path)
    img = (img / 255.).astype(np.float32)
    img = cv2.resize(img, dsize = (224,224))
    embedding_vector = vgg_face_descriptor.predict(np.expand_dims(img, axis=0))[0]
    embeddings[i]=embedding_vector
```

```
1/1 [==============================] - 0s 398ms/step
1/1 [==============================] - 0s 399ms/step
1/1 [==============================] - 0s 406ms/step
```

- ❖ Ensure you provide the correct path to the image to be recognized in the code.

Predicting labels for celebrity images sourced from the web whilst their label incorporated in our dataset.

```
[ ]  img_path = '/kaggle/input/testimage/test.jpg'
     img = load_image(img_path)

     img = (img / 255.).astype(np.float32)
     img = cv2.resize(img, dsize = (224,224))
     print(img.shape)

     embedding_vector1 = vgg_face_descriptor.predict(np.expand_dims(img, axis=0))[0]
     print(embedding_vector1.shape)
```