Ciphense Project Assignment

Question- 1

Write a REST API: "/getSnapInfo" POST type. It takes a picture and uses a computer vision/image recognition api such as google or microsoft and determines all the details that are there in the picture as recognized by the called api. It then translates the content returned to speech and returns back such information as text as well as a speech file such as mp3 or avi.

Python Libraries Used

- gTTs
- requests
- urllib
- json
- os

Microsoft Azure API Linking

- A free Azure account was created and the subscription keys were obtained
- A locally stored image to extract visual features by using Computer Vision's REST API. With the analyze image method, we can extract visual features based on image content.
- The vision_base_url is set to the version required, here it’s 2.0.
- analyze_url = vision_base_url + "analyze“

  This step completes the preprocessing phase.

How the code works??

- The required image of code has been entered in the code.
- The required request parameters are entered into a variable ‘params’. Here only visual features have been included.
- The visual features include Categories,Description,Color,ImageType,Objects,Tags,Adult,Brands,Faces.
- response.raise_for_status(). This line sends the image to azure server for all the analysis we require and the output is stored in a response.json file.
- The json file is converted to a suitable string using ‘dumps’ and ‘loads’ functionality of the json package.

Sample output of json file(only a part has been displayed here)

```
'categories': [
  {
    'name': 'people_group',
    'score': 0.6953125,
    'detail': {
      'celebrities': [
        {
          'name': 'Anil Kapoor',
          'confidence': 0.9999927282333374,
          'faceRectangle': {
            'left': 306,
            'top': 212,
            'width': 67,
            'height': 67
          }
        },
        {
          'name': 'Tom Cruise',
          'confidence': 0.9934597015380859,
          'faceRectangle': {
            'left': 435,
            'top': 214,
            'width': 66,
            'height': 66
          }
        }
      ]
    }
  }
```

How the code works(contd..)

- The formatted json file is stored in an ‘analyze’ variable.
- This variable is a dictionary which has all the lists like ‘categories’, ‘description’, ‘tags’, etc.
- Now the ‘analyze’ variable’s lists are checked and iterated respectively to get multiple tags or faces, etc if available. While we keep doing this iteration we store them in a string variable with “\n” as the escape sequence so that we can finally write the string to a text file.

How the code works(contd..)

- The formatted json file is stored in an ‘analyze’ variable.
- This variable is a dictionary which has all the lists like ‘categories’, ‘description’, ‘tags’, etc.
- Now the ‘analyze’ variable’s lists are checked and iterated respectively to get multiple tags or faces, etc if available. While we keep doing this iteration we store them in a string variable with “\n” as the escape sequence so that we can finally write the string to a text file.

Use of gTTs

- gTTs (Google speech to text Python)
- Here we had 2 options, either read sequentially from the created text file and keep on adding to the mp3 file, or directly use the string variable in which the total contents of file was stored.
- I used the string variable as it looked easy.
- myobj = gTTS(text=ans, lang=language, slow=False)

  myobj.save(“welcome.mp3")

  These 2 lines take the string ‘ans’ as the i/p and use the gTTs object to create a mp3 file in the local directory in which the source code has been saved.