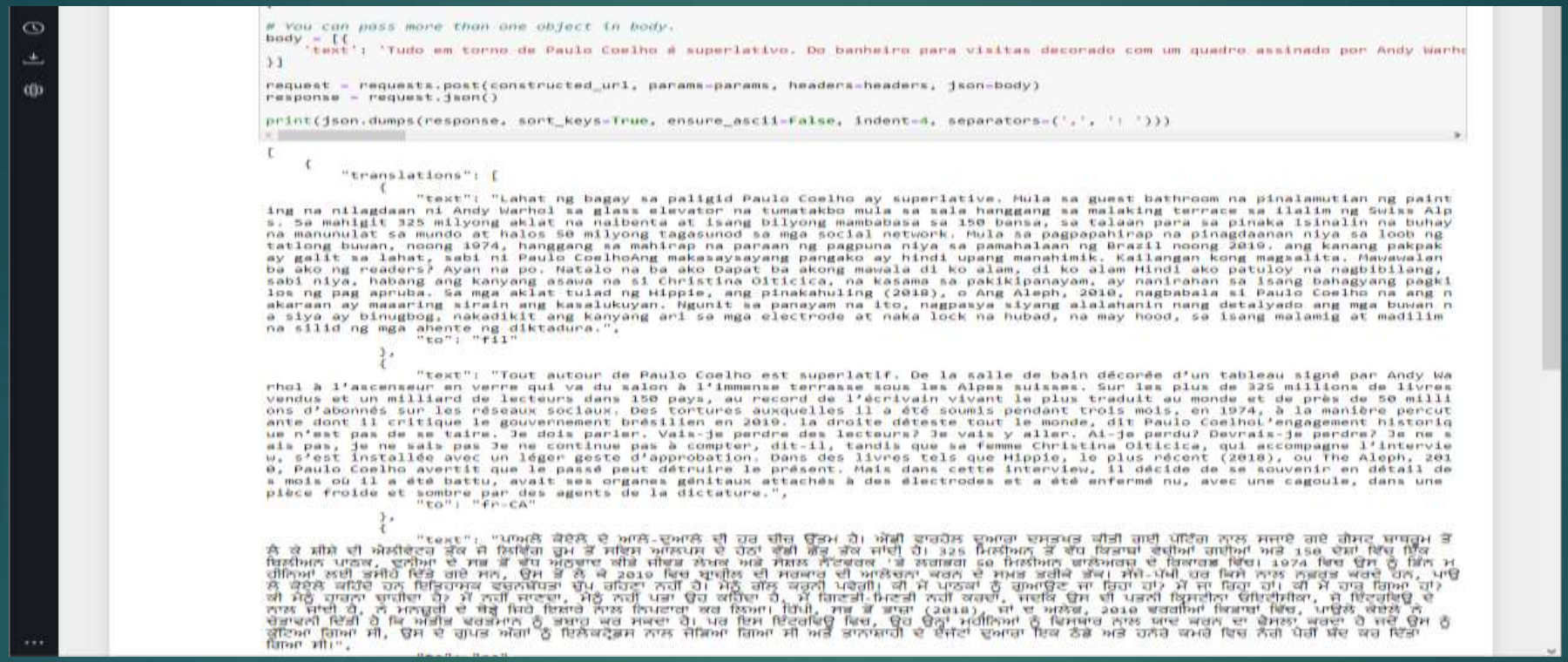
Challenge 4 -- Part 2

Azure Cognitive Services

AI Infrastructure and Arch.

SAMRAT THAPA -- 200481763

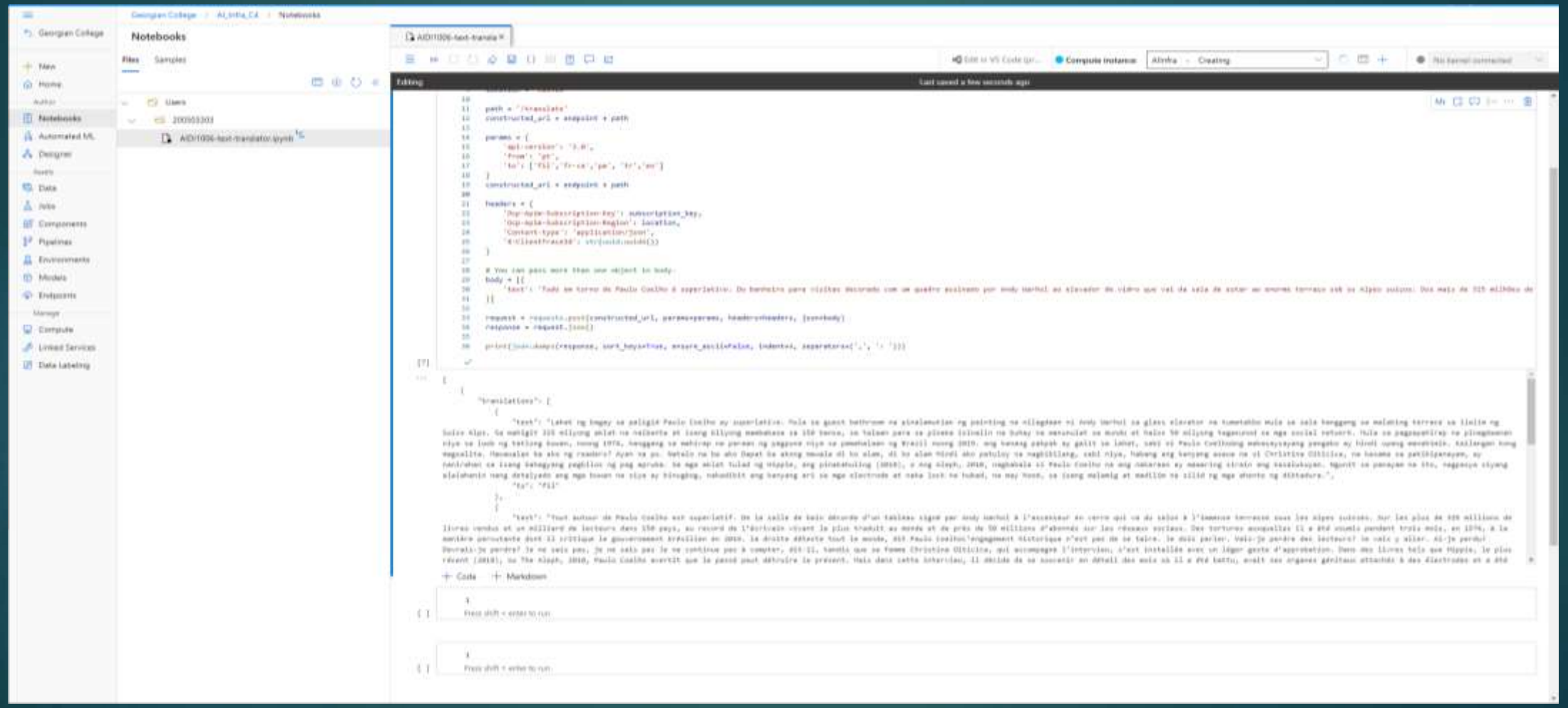
Challenge 4.6



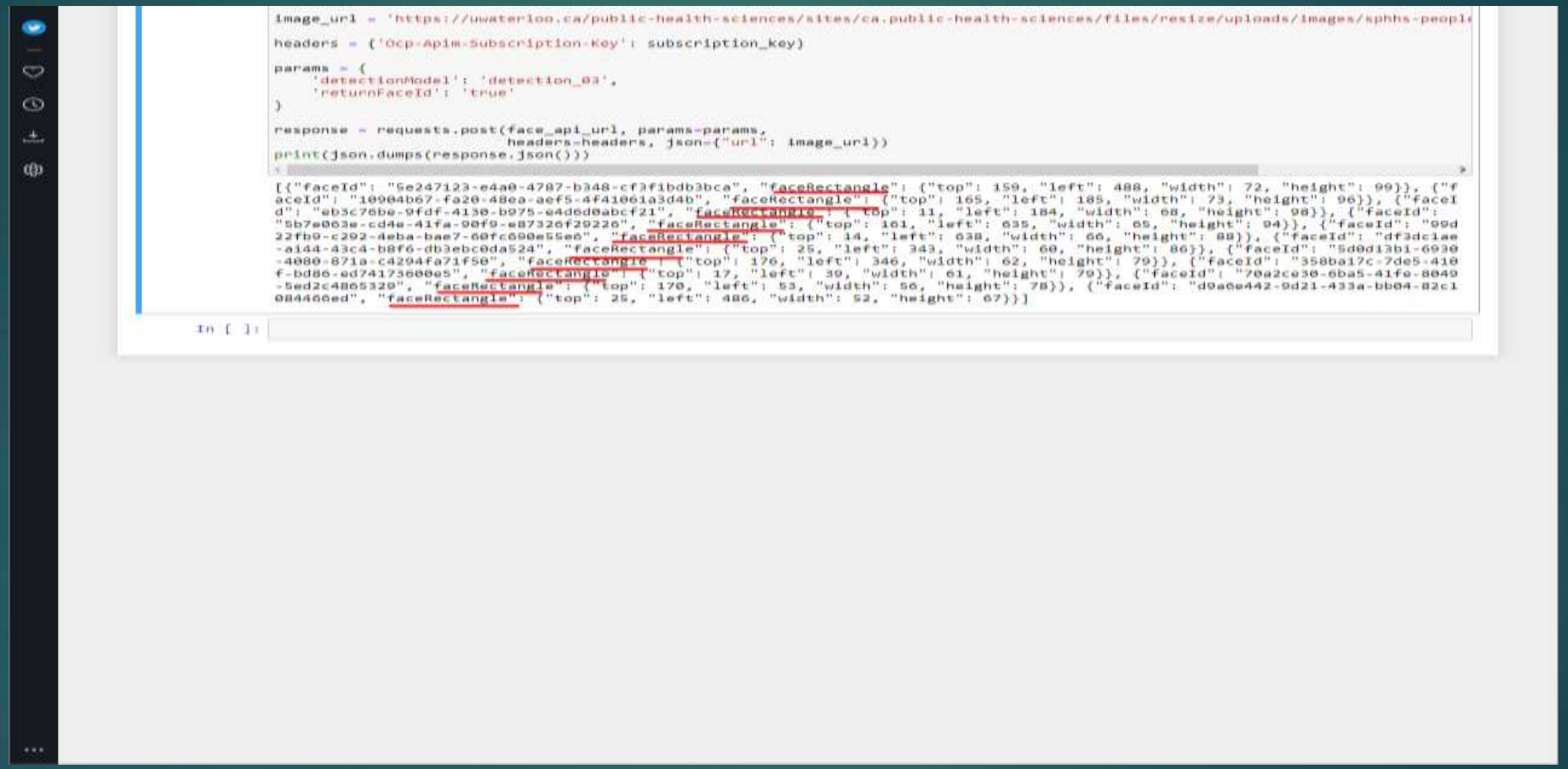
```
# You can pass more than one object in body.
body = [{
    'text': 'Tudo em torno de Paulo Coelho é superlativo. Do banheiro para visitas decorado com um quadro assinado por Andy Warh
}]

request = requests.post(constructed_url, params=params, headers=headers, json=body)
response = request.json()

print(json.dumps(response, sort_keys=True, ensure_ascii=False, indent=4, separators=(',', ': ')))
```


```
[
    {
        "translations": [
            {
                "text": "Lahat ng bagay sa paligid Paulo Coelho ay superlative. Mula sa guest bathroom na pinalamutian ng paint
ing na nilagdaan ni Andy Warhol sa glass elevator na tumatakbo mula sa sala hanggang sa malaking terrace sa ilalim ng Swiss Alp
s. Sa mahigit 325 milyong aklat na naibenta at isang bilyong mambabasa sa 150 bansa, sa talaan para sa pinaka isinalin na buhay
na manunulat sa mundo at halos 50 milyong tagasunod sa mga social network. Mula sa pagpapahirap na pinagdaanan niya sa loob ng
tatlong buwan, noong 1974, hanggang sa mahirap na paraan ng pagpuna niya sa pamahalaan ng Brazil noong 2019. ang kanang pakpak
ay galit sa lahat, sabi ni Paulo CoelhoAng makasaysayang pangako ay hindi upang manahimik. Kailangan kong magsalita. Mawawalan
ba ako ng readers? Ayan na po. Natalo na ba ako Dapat ba akong mawala di ko alam, di ko alam Hindi ako patuloy na nagbibilang,
sabi niya, habang ang kanyang asawa na si Christina Oiticica, na kasama sa pakikipanayam, ay nanirahan sa isang bahagyang pagki
los ng pag apruba. Sa mga aklat tulad ng Hippie, ang pinakahuling (2018), o Ang Aleph, 2010, nagbabala si Paulo Coelho na ang n
akaraan ay maaaring sirain ang kasalukuyan. Ngunit sa panayam na ito, nagpasya siyang alalahanin nang detalyado ang mga buwan n
a siya ay binugbog, nakadikit ang kanyang ari sa mga electrode at naka lock na hubad, na may hood, sa isang malamig at madilim
na silid ng mga ahente ng diktadura.",
                "to": "fil"
            },
            {
                "text": "Tout autour de Paulo Coelho est superlatif. De la salle de bain décorée d'un tableau signé par Andy Wa
rhol à l'ascenseur en verre qui va du salon à l'immense terrasse sous les Alpes suisses. Sur les plus de 325 millions de livres
vendus et un milliard de lecteurs dans 150 pays, au record de l'écrivain vivant le plus traduit au monde et de près de 50 milli
ons d'abonnés sur les réseaux sociaux. Des tortures auxquelles il a été soumis pendant trois mois, en 1974, à la manière percut
ante dont il critique le gouvernement brésilien en 2019. la droite déteste tout le monde, dit Paulo Coelhol'engagement historiq
ue n'est pas de se taire. Je dois parler. Vais-je perdre des lecteurs? Je vais y aller. Ai-je perdu? Devrais-je perdre? Je ne s
ais pas, je ne sais pas Je ne continue pas à compter, dit-il, tandis que sa femme Christina Oiticica, qui accompagne l'intervie
w, s'est installée avec un léger geste d'approbation. Dans des livres tels que Hippie, le plus récent (2018), ou The Aleph, 201
0, Paulo Coelho avertit que le passé peut détruire le présent. Mais dans cette interview, il décide de se souvenir en détail de
s mois où il a été battu, avait ses organes génitaux attachés à des électrodes et a été enfermé nu, avec une cagoule, dans une
pièce froide et sombre par des agents de la dictature.",
                "to": "fr-CA"
            },
            {
                "text": "ਪਾਓਲੋ ਕੋਏਲੋ ਦੇ ਆਲੇ-ਦੁਆਲੇ ਦੀ ਹਰ ਚੀਜ਼ ਉੱਤਮ ਹੈ। ਐਂਡੀ ਵਾਰਹੋਲ ਦੁਆਰਾ ਦਸਤਖਤ ਕੀਤੀ ਗਈ ਪੇਂਟਿੰਗ ਨਾਲ ਸਜਾਏ ਗਏ ਗੈਸਟ ਬਾਥਰੂਮ ਤੋਂ
ਲੈ ਕੇ ਸ਼ੀਸ਼ੇ ਦੀ ਐਲੀਵੇਟਰ ਤੱਕ ਜੋ ਸਿਖਿਰ ਪੂਮ ਤੋਂ ਸਵਿਸ ਆਲਪਸ ਦੇ ਹੇਠਾਂ ਵੱਡੀ ਛੱਤ ਤੱਕ ਜਾਂਦੀ ਹੈ। 325 ਮਿਲੀਅਨ ਤੋਂ ਵੱਧ ਕਿਤਾਬਾਂ ਵਿਕੀਆਂ ਗਈਆਂ ਅਤੇ 150 ਦੇਸ਼ਾਂ ਵਿਚ ਇੱਕ
ਬਿਲੀਅਨ ਪਾਠਕ, ਦੁਨੀਆ ਦੇ ਸਭ ਤੋਂ ਵੱਧ ਅਨੁਵਾਦ ਕੀਤੇ ਜੀਵਤ ਲੇਖਕ ਅਤੇ ਸੋਸ਼ਲ ਨੈੱਟਵਰਕ 'ਤੇ ਲਗਭਗ 50 ਮਿਲੀਅਨ ਫਾਲੋਅਰਜ਼ ਦੇ ਰਿਕਾਰਡ ਵਿਚ। 1974 ਵਿਚ ਉਸ ਨੂੰ ਤਿੰਨ ਮ
ਹੀਨਿਆਂ ਲਈ ਤਸੀਹੇ ਦਿੱਤੇ ਗਏ ਸਨ, ਉਸ ਤੋਂ ਲੈ ਕੇ 2019 ਵਿਚ ਬ੍ਰਾਜ਼ੀਲ ਦੀ ਸਰਕਾਰ ਦੀ ਆਲੋਚਨਾ ਕਰਨ ਦੇ ਸਮਤ ਤਰੀਕੇ ਤੱਕ। ਸੱਜੇ-ਪੱਖੀ ਹਰ ਕਿਸੇ ਨਾਲ ਨਫ਼ਰਤ ਕਰਦੇ ਹਨ, ਪਾਉ
ਲੋ ਕੋਏਲੋ ਕਹਿੰਦੇ ਹਨ ਇਤਿਹਾਸਕ ਦਸਤਖੰਡਤਾ ਚੁੱਪ ਰਹਿਣਾ ਨਹੀਂ ਹੈ। ਮੈਨੂੰ ਗੱਲ ਕਰਨੀ ਪਵੇਗੀ। ਕੀ ਮੈਂ ਪਾਠਕਾਂ ਨੂੰ ਗੁਆਉਣ ਜਾ ਰਿਹਾ ਹਾਂ? ਮੈਂ ਜਾ ਰਿਹਾ ਹਾਂ। ਕੀ ਮੈਂ ਹਾਰ ਗਿਆ ਹਾਂ?
ਕੀ ਮੈਨੂੰ ਹਾਰਨਾ ਚਾਹੀਦਾ ਹੈ? ਮੈਂ ਨਹੀਂ ਜਾਣਦਾ, ਮੈਨੂੰ ਨਹੀਂ ਪਤਾ ਉਹ ਕਹਿਆ ਹੈ, ਮੈਂ ਗਿਣਤੀ-ਮਿਣਤੀ ਨਹੀਂ ਕਰਦੀ, ਸਵਰਿ ਉਸ ਦੀ ਪਤਨੀ ਕ੍ਰਿਸਟੀਨਾ ਓਇਟੀਸੀਕਾ, ਜੋ ਇੰਟਰਵਿਊ ਦੇ
ਨਾਲ ਜਾਂਦੀ ਹੈ, ਨੇ ਸਨਚੁਰੀ ਦੇ ਥੋੜ੍ਹੇ ਜਿਹੇ ਇਸ਼ਾਰੇ ਨਾਲ ਸਿਪਾਟਾ ਕਰ ਲਿਆ। ਹਿੱਪੀ, ਸਭ ਤੋਂ ਫ਼ਾਲਾ (2018), ਜਾ ਦ ਅਲੇਫ਼, 2010 ਵਰਗੀਆਂ ਕਿਤਾਬਾਂ ਵਿਚ, ਪਾਉਲੋ ਕੋਏਲੋ ਨੇ
ਚੇਤਾਵਨੀ ਦਿੱਤੀ ਹੈ ਕਿ ਅਤੀਤ ਵਰਤਮਾਨ ਨੂੰ ਤਬਾਹ ਕਰ ਸਕਦਾ ਹੈ। ਪਰ ਇਸ ਇੰਟਰਵਿਊ ਵਿਚ, ਉਹ ਉਨ੍ਹਾਂ ਮਹੀਨਿਆਂ ਨੂੰ ਵਿਸਥਾਰ ਨਾਲ ਯਾਦ ਕਰਨ ਦਾ ਫੈਸਲਾ ਕਰਦਾ ਹੈ ਜਦੋਂ ਉਸ ਨੂੰ
ਕੁੱਟਿਆ ਗਿਆ ਸੀ, ਉਸ ਦੇ ਗੁਪਤ ਅੰਗਾਂ ਨੂੰ ਇਲੈਕਟ੍ਰੋਡਸ ਨਾਲ ਜੋੜਿਆ ਗਿਆ ਸੀ ਅਤੇ ਡਾਕਟਾਰੀ ਦੇ ਏਜੰਟਾਂ ਦੁਆਰਾ ਇਕ ਠੰਡੇ ਅਤੇ ਹਨੇਰੇ ਕਮਰੇ ਵਿਚ ਨੰਗੇ ਪੈਰੀਂ ਬੰਦ ਕਰ ਦਿੱਤਾ
ਗਿਆ ਸੀ।",
```

Challenge 4.7



Challenge 4.8

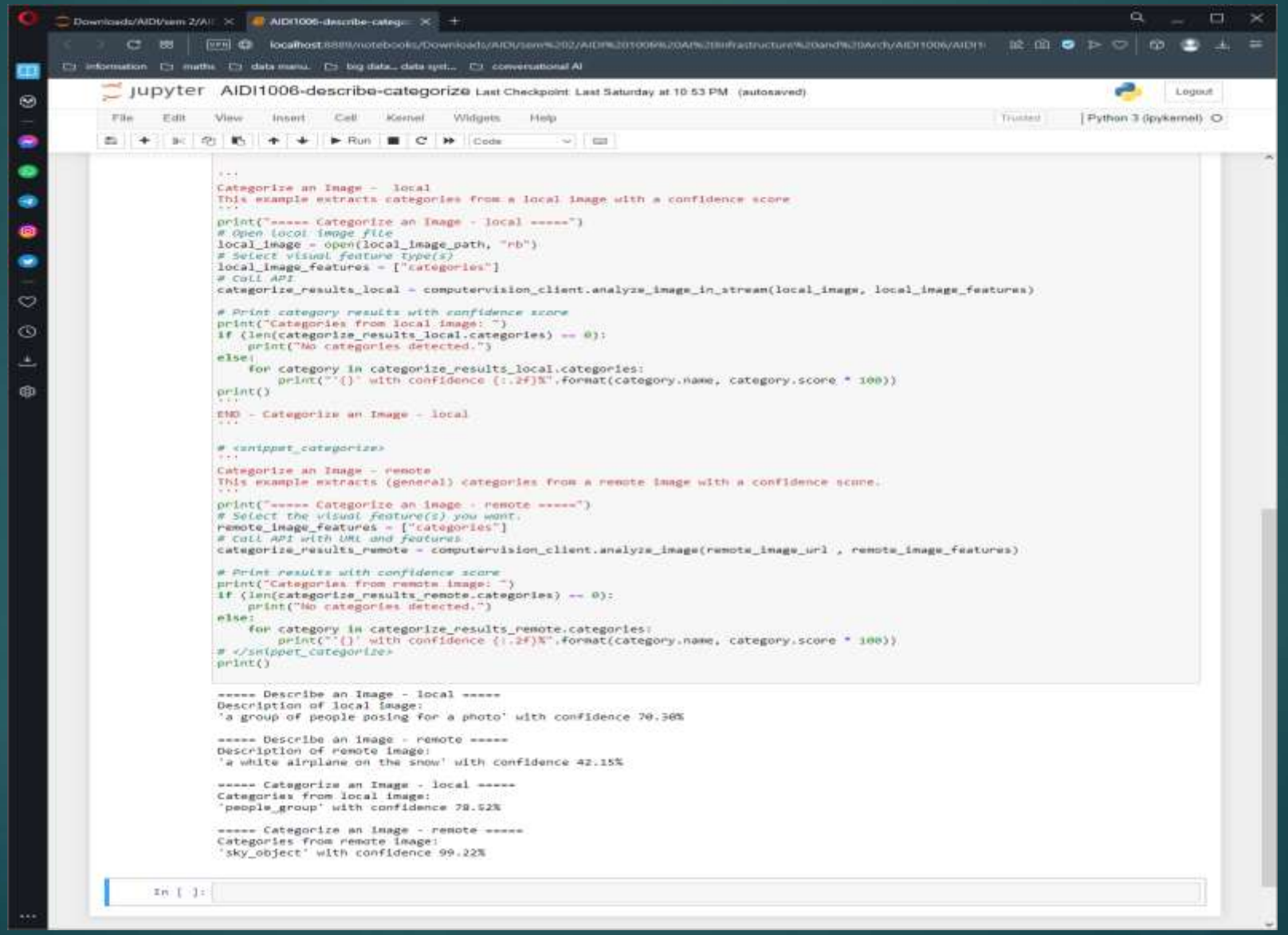


```
image_url = 'https://uwaterloo.ca/public-health-sciences/sites/ca.public-health-sciences/files/resize/uploads/images/sphhs-people

headers = {'Ocp-Apim-Subscription-Key': subscription_key}

params = {
    'detectionModel': 'detection_03',
    'returnFaceId': 'true'
}

response = requests.post(face_api_url, params=params,
                         headers=headers, json={"url": image_url})
print(json.dumps(response.json()))
```

```
[{"faceId": "5e247123-e4a0-4787-b348-cf3f1bdb3bca", "faceRectangle": {"top": 159, "left": 488, "width": 72, "height": 99}}, {"f
aceId": "10904b67-fa20-48ea-aef5-4f41061a3d4b", "faceRectangle": {"top": 165, "left": 185, "width": 73, "height": 96}}, {"faceI
d": "eb3c76be-9fdf-4130-b975-e4d6d0abcf21", "faceRectangle": {"top": 11, "left": 184, "width": 68, "height": 98}}, {"faceId":
"5b7e063e-cd4e-41fa-90f9-e87326f29226", "faceRectangle": {"top": 161, "left": 635, "width": 65, "height": 94}}, {"faceId": "99d
22fb9-c292-4eba-bae7-60fc690e55e6", "faceRectangle": {"top": 14, "left": 638, "width": 66, "height": 88}}, {"faceId": "df3dc1ae
-a144-43c4-b8f6-db3ebc0da524", "faceRectangle": {"top": 25, "left": 343, "width": 60, "height": 86}}, {"faceId": "5d0d13b1-6930
-4080-871a-c4294fa71f50", "faceRectangle": {"top": 176, "left": 346, "width": 62, "height": 79}}, {"faceId": "358ba17c-7de5-410
f-bd86-ed7417c600e5", "faceRectangle": {"top": 17, "left": 39, "width": 61, "height": 79}}, {"faceId": "70a2ce30-6ba5-41f0-8049
-5ed2c4865329", "faceRectangle": {"top": 170, "left": 53, "width": 56, "height": 78}}, {"faceId": "d9a6e442-9d21-433a-bb04-82c1
084466ed", "faceRectangle": {"top": 25, "left": 486, "width": 52, "height": 67}}]
```

```
In [ ]:
```

Challenge 4.9

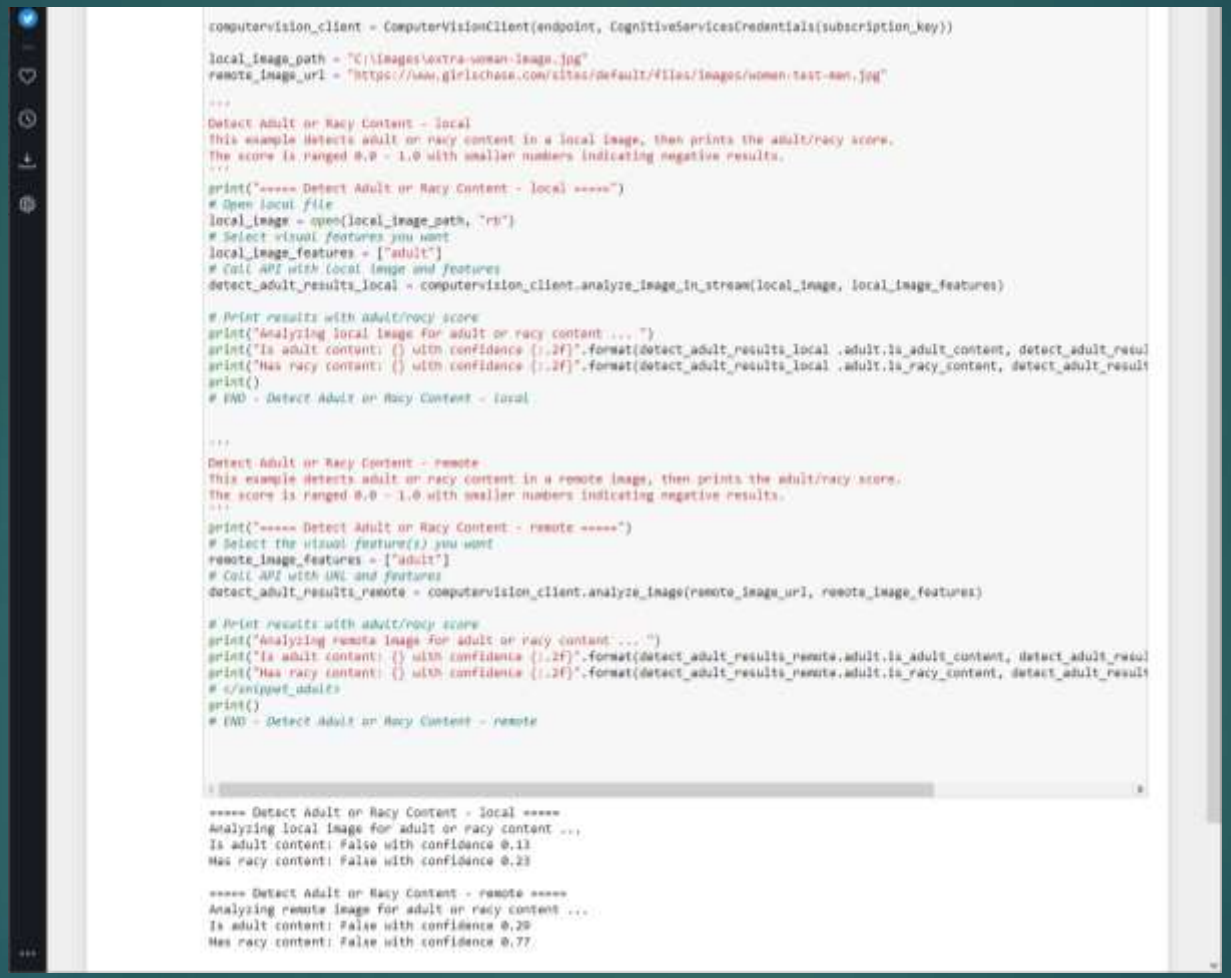


```
computervision_client = ComputerVisionClient(endpoint, CognitiveServicesCredentials(subscription_key))
remote_image_url = "https://wp.en.alete1a.org/wp-content/uploads/sites/2/2020/05/web3-family-large-big-home-brother-sister-mother
# local_file = "C:\\images\\happy-family.jpg"
# Tag an image - remote // This example returns a tag (key word) for each thing in the image.
print("===== Tag an image - remote =====")
# Call API with remote image
tags_result_remote = computervision_client.tag_image(remote_image_url)

# Print results with confidence score
print("Tags in the remote image: ")
if (len(tags_result_remote.tags) == 0):
    print("No tags detected.")
else:
    for tag in tags_result_remote.tags:
        print("'{}' with confidence {:.2f}%".format(tag.name, tag.confidence * 100))
```

```
===== Tag an image - remote =====
Tags in the remote image:
'clothing' with confidence 99.93%
'person' with confidence 99.89%
'human face' with confidence 99.79%
'smile' with confidence 98.06%
'girl' with confidence 94.15%
'indoor' with confidence 94.03%
'people' with confidence 91.98%
'couch' with confidence 91.41%
'lap' with confidence 91.19%
'social group' with confidence 90.77%
'jeans' with confidence 90.46%
'family' with confidence 89.42%
'group' with confidence 88.85%
'wall' with confidence 87.76%
'friendship' with confidence 87.64%
'comfort' with confidence 87.07%
'sitting' with confidence 83.09%
'woman' with confidence 80.23%
'posing' with confidence 69.20%
```

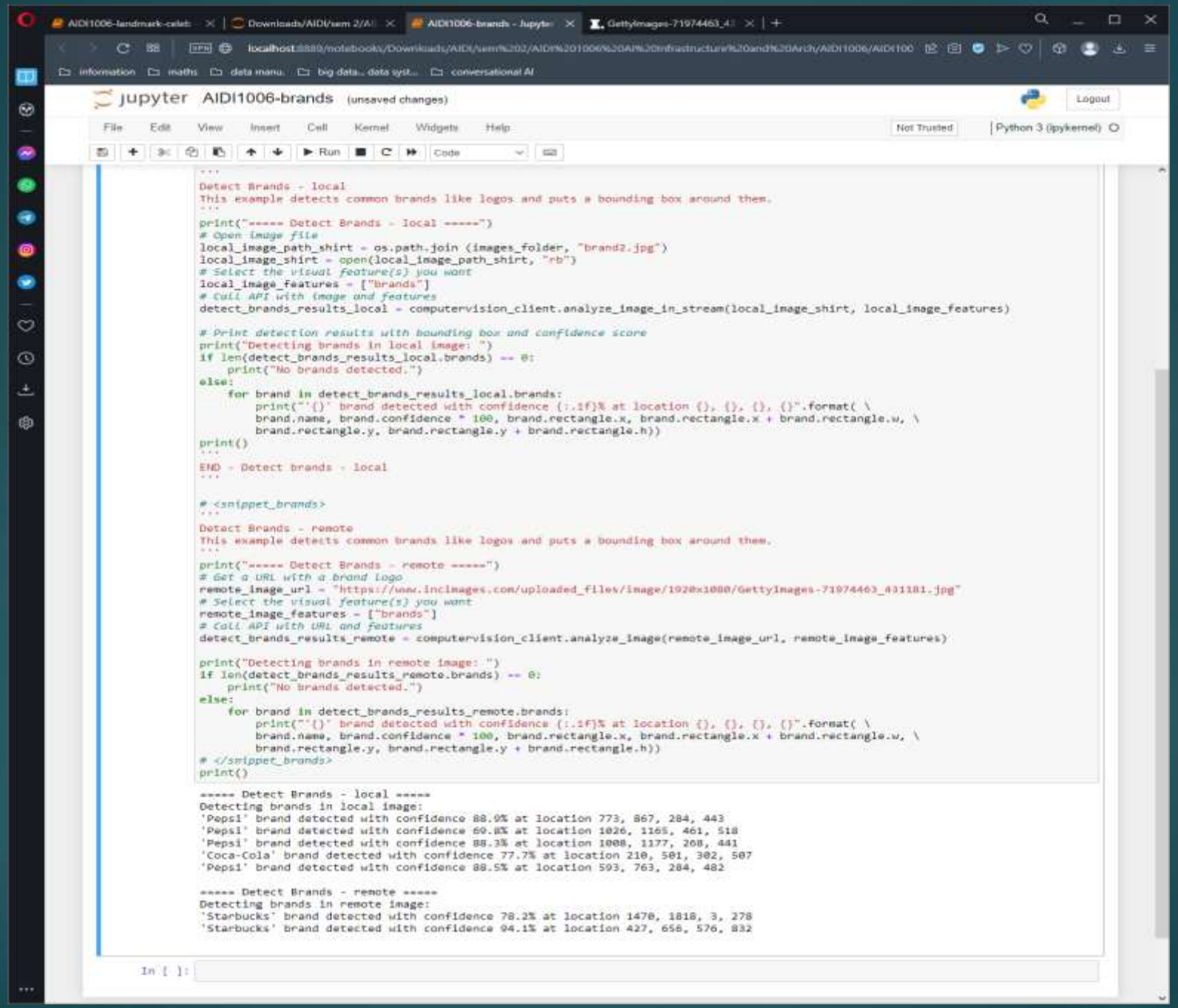
In [ ]:

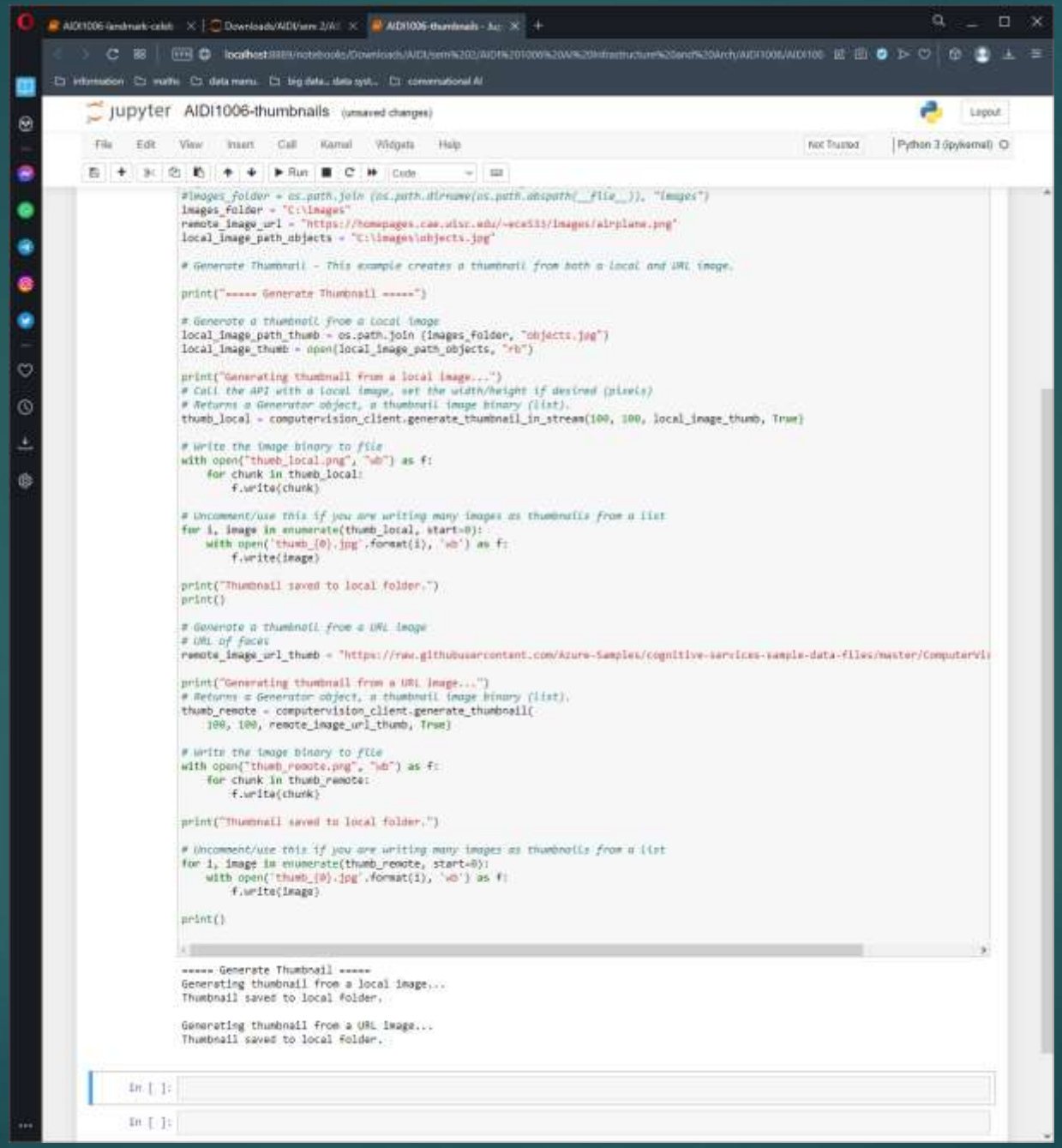
In [ ]:

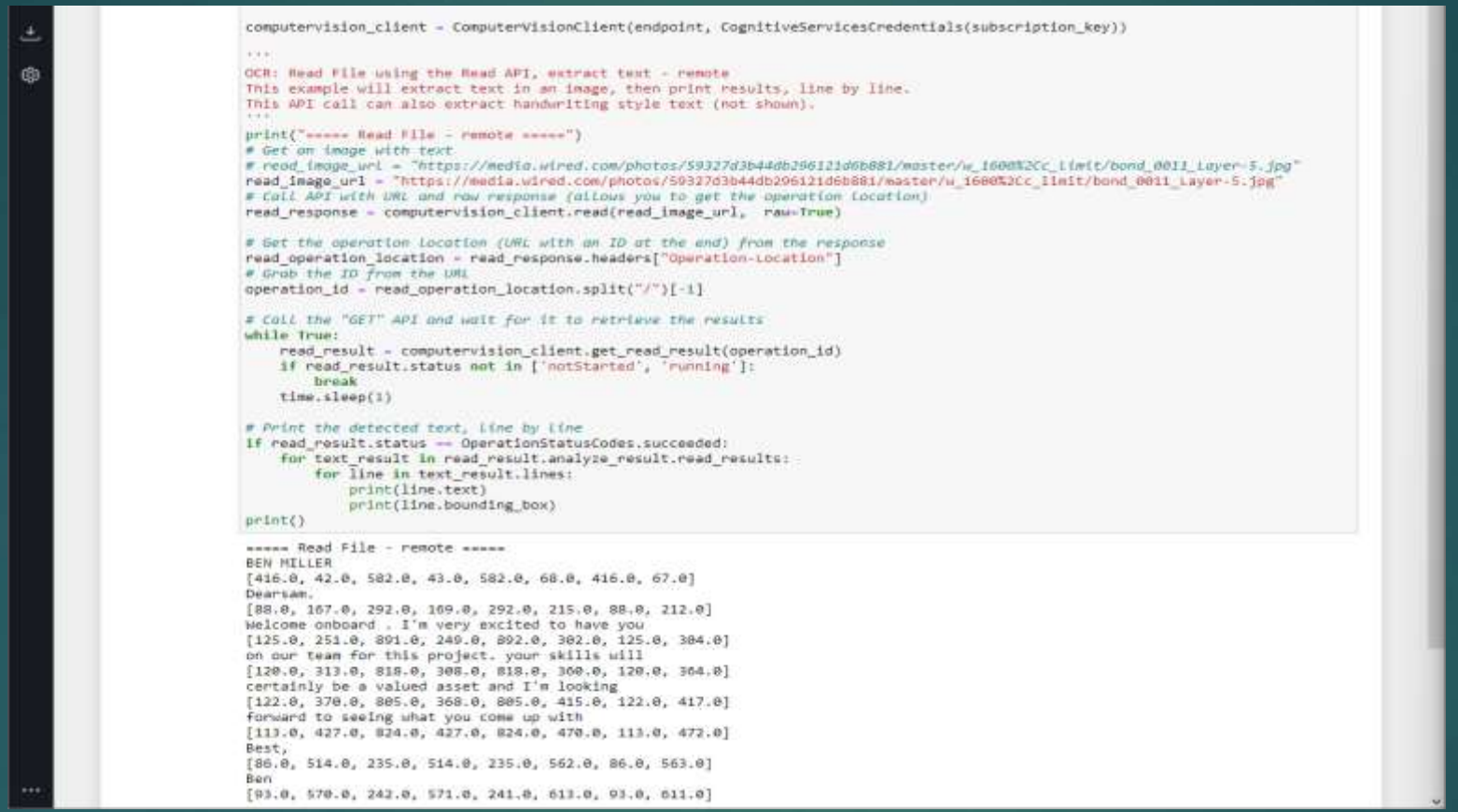


```
computervision_client = ComputerVisionClient(endpoint, CognitiveServicesCredentials(subscription_key))

local_image_path = "C:\Images\extra-woman-image.jpg"
remote_image_url = "https://www.girlschase.com/sites/default/files/images/women-test-men.jpg"

'''
Detect Adult or Racy Content - local
This example detects adult or racy content in a local image, then prints the adult/racy score.
The score is ranged 0.0 - 1.0 with smaller numbers indicating negative results.
'''
print("===== Detect Adult or Racy Content - local =====")
# Open local file
local_image = open(local_image_path, "rb")
# Select visual features you want
local_image_features = ["adult"]
# Call API with local image and features
detect_adult_results_local = computervision_client.analyze_image_in_stream(local_image, local_image_features)

# Print results with adult/racy score
print("Analyzing local image for adult or racy content ... ")
print("Is adult content: {} with confidence {:.2f}".format(detect_adult_results_local .adult.is_adult_content, detect_adult_resul
print("Has racy content: {} with confidence {:.2f}".format(detect_adult_results_local .adult.is_racy_content, detect_adult_result
print()
# END - Detect Adult or Racy Content - local


'''
Detect Adult or Racy Content - remote
This example detects adult or racy content in a remote image, then prints the adult/racy score.
The score is ranged 0.0 - 1.0 with smaller numbers indicating negative results.
'''
print("===== Detect Adult or Racy Content - remote =====")
# Select the visual feature(s) you want
remote_image_features = ["adult"]
# Call API with URL and features
detect_adult_results_remote = computervision_client.analyze_image(remote_image_url, remote_image_features)

# Print results with adult/racy score
print("Analyzing remote image for adult or racy content ... ")
print("Is adult content: {} with confidence {:.2f}".format(detect_adult_results_remote.adult.is_adult_content, detect_adult_resul
print("Has racy content: {} with confidence {:.2f}".format(detect_adult_results_remote.adult.is_racy_content, detect_adult_result
# </snippet_adult>
print()
# END - Detect Adult or Racy Content - remote
```

```
===== Detect Adult or Racy Content - local =====
Analyzing local image for adult or racy content ...
Is adult content: False with confidence 0.13
Has racy content: False with confidence 0.23

===== Detect Adult or Racy Content - remote =====
Analyzing remote image for adult or racy content ...
Is adult content: False with confidence 0.29
Has racy content: False with confidence 0.77
```

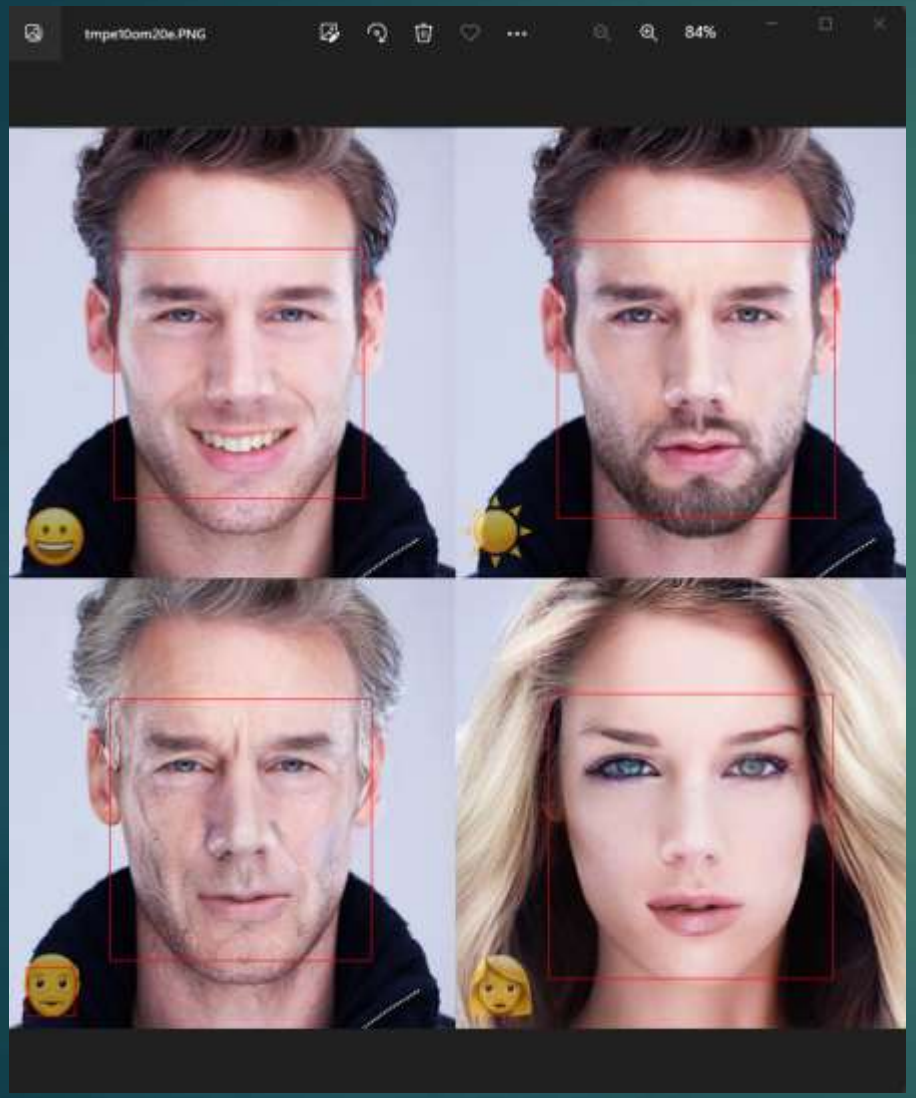

AIDI1006-landmark-celeb
Downloads/AIDI/sem 2/AI
AIDI1006-brands - Jupyter
GettyImages-71974463_4
localhost:8889/notebooks/Downloads/AIDI/sem%202/AIDI%201006%20AI%20Infrastructure%20and%20Arch/AIDI1006/AIDI100
information
maths
data manu.
big data.. data syst...
conversational AI
jupyter AIDI1006-brands (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
'''
Detect Brands - local
This example detects common brands like logos and puts a bounding box around them.
'''
print("===== Detect Brands - local =====")
# Open image file
local_image_path_shirt = os.path.join (images_folder, "brand2.jpg")
local_image_shirt = open(local_image_path_shirt, "rb")
# Select the visual feature(s) you want
local_image_features = ["brands"]
# Call API with image and features
detect_brands_results_local = computervision_client.analyze_image_in_stream(local_image_shirt, local_image_features)
# Print detection results with bounding box and confidence score
print("Detecting brands in local image: ")
if len(detect_brands_results_local.brands) == 0:
print("No brands detected.")
else:
for brand in detect_brands_results_local.brands:
print("'{}' brand detected with confidence {:.1f}% at location {}, {}, {}, {}".format(\
brand.name, brand.confidence * 100, brand.rectangle.x, brand.rectangle.x + brand.rectangle.w, \
brand.rectangle.y, brand.rectangle.y + brand.rectangle.h))
print()
'''
END - Detect brands - local
'''
<snippet_brands>
'''
Detect Brands - remote
This example detects common brands like logos and puts a bounding box around them.
'''
print("===== Detect Brands - remote =====")
# Get a URL with a brand logo
remote_image_url = "https://www.inclmages.com/uploaded_files/image/1920x1080/GettyImages-71974463_431181.jpg"
# Select the visual feature(s) you want
remote_image_features = ["brands"]
# Call API with URL and features
detect_brands_results_remote = computervision_client.analyze_image(remote_image_url, remote_image_features)
print("Detecting brands in remote image: ")
if len(detect_brands_results_remote.brands) == 0:
print("No brands detected.")
else:
for brand in detect_brands_results_remote.brands:
print("'{}' brand detected with confidence {:.1f}% at location {}, {}, {}, {}".format(\
brand.name, brand.confidence * 100, brand.rectangle.x, brand.rectangle.x + brand.rectangle.w, \
brand.rectangle.y, brand.rectangle.y + brand.rectangle.h))
# </snippet_brands>
print()
===== Detect Brands - local =====
Detecting brands in local image:
'Pepsi' brand detected with confidence 88.9% at location 773, 867, 284, 443
'Pepsi' brand detected with confidence 60.8% at location 1026, 1165, 461, 518
'Pepsi' brand detected with confidence 88.3% at location 1008, 1177, 268, 441
'Coca-Cola' brand detected with confidence 77.7% at location 210, 501, 302, 507
'Pepsi' brand detected with confidence 88.5% at location 593, 763, 284, 482
===== Detect Brands - remote =====
Detecting brands in remote image:
'Starbucks' brand detected with confidence 78.2% at location 1470, 1818, 3, 278
'Starbucks' brand detected with confidence 94.1% at location 427, 656, 576, 832
In [ ]:

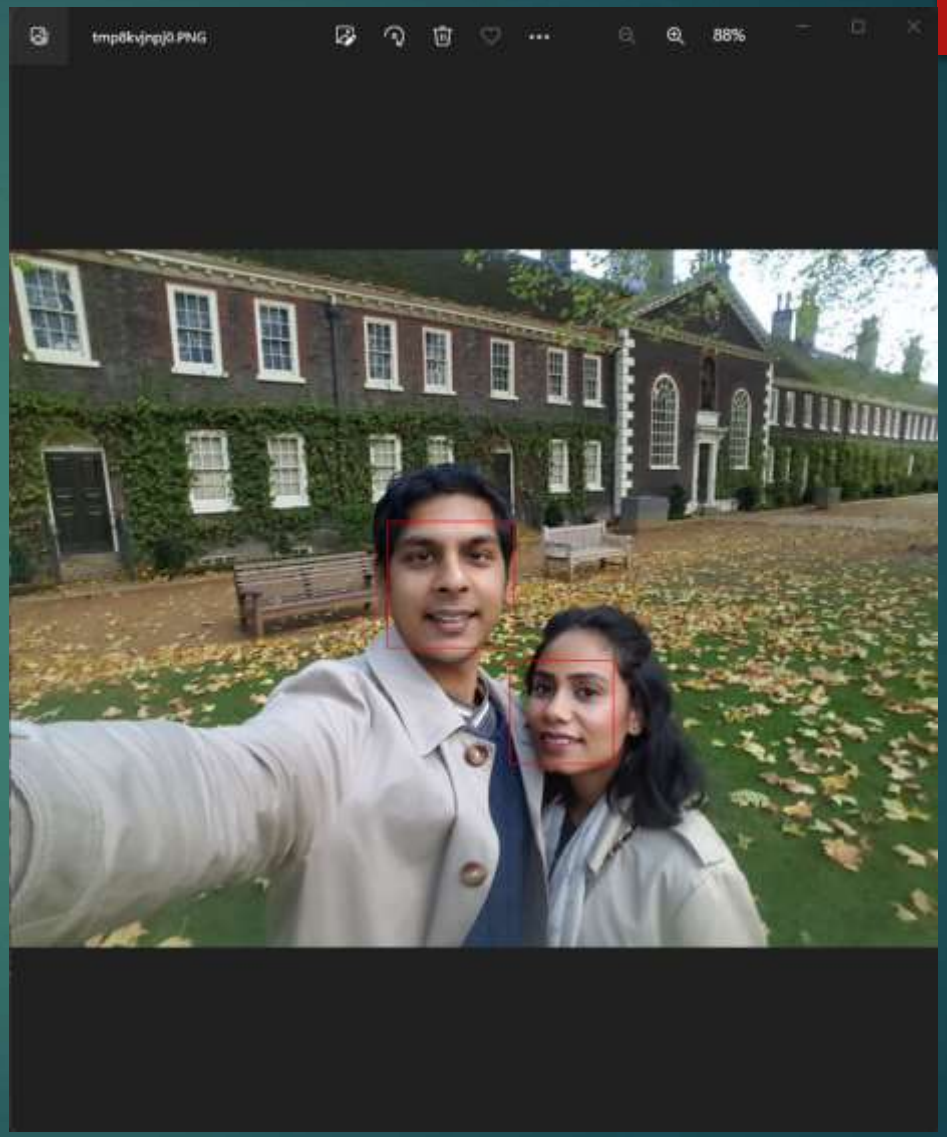

jupyter AIDI1006-thumbnails (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
#images_folder = os.path.join (os.path.dirname(os.path.abspath(__file__)), "images")
images_folder = "C:\Images"
remote_image_url = "https://homepages.cae.wisc.edu/~ece533/images/airplane.png"
local_image_path_objects = "C:\Images\objects.jpg"
# Generate Thumbnail - This example creates a thumbnail from both a local and URL image.
print("----- Generate Thumbnail -----")
# Generate a thumbnail from a local image
local_image_path_thumb = os.path.join (images_folder, "objects.jpg")
local_image_thumb = open(local_image_path_objects, "rb")
print("Generating thumbnail from a local image...")
# Call the API with a local image, set the width/height if desired (pixels)
# Returns a Generator object, a thumbnail image binary (list).
thumb_local = computervision_client.generate_thumbnail_in_stream(100, 100, local_image_thumb, True)
# Write the image binary to file
with open("thumb_local.png", "wb") as f:
for chunk in thumb_local:
f.write(chunk)
# Uncomment/use this if you are writing many images as thumbnails from a list
for i, image in enumerate(thumb_local, start=0):
with open('thumb_{0}.jpg'.format(i), 'wb') as f:
f.write(image)
print("Thumbnail saved to local folder.")
print()
# Generate a thumbnail from a URL image
# URL of faces
remote_image_url_thumb = "https://raw.githubusercontent.com/Azure-Samples/cognitive-services-sample-data-files/master/ComputerVi
print("Generating thumbnail from a URL image...")
# Returns a Generator object, a thumbnail image binary (list).
thumb_remote = computervision_client.generate_thumbnail(
100, 100, remote_image_url_thumb, True)
Write the image binary to file
with open("thumb_remote.png", "wb") as f:
for chunk in thumb_remote:
f.write(chunk)
print("Thumbnail saved to local folder.")
# Uncomment/use this if you are writing many images as thumbnails from a list
for i, image in enumerate(thumb_remote, start=0):
with open('thumb_{0}.jpg'.format(i), 'wb') as f:
f.write(image)
print()
----- Generate Thumbnail -----
Generating thumbnail from a local image...
Thumbnail saved to local folder.
Generating thumbnail from a URL image...
Thumbnail saved to local folder.
In [ ]:
In [ ]:

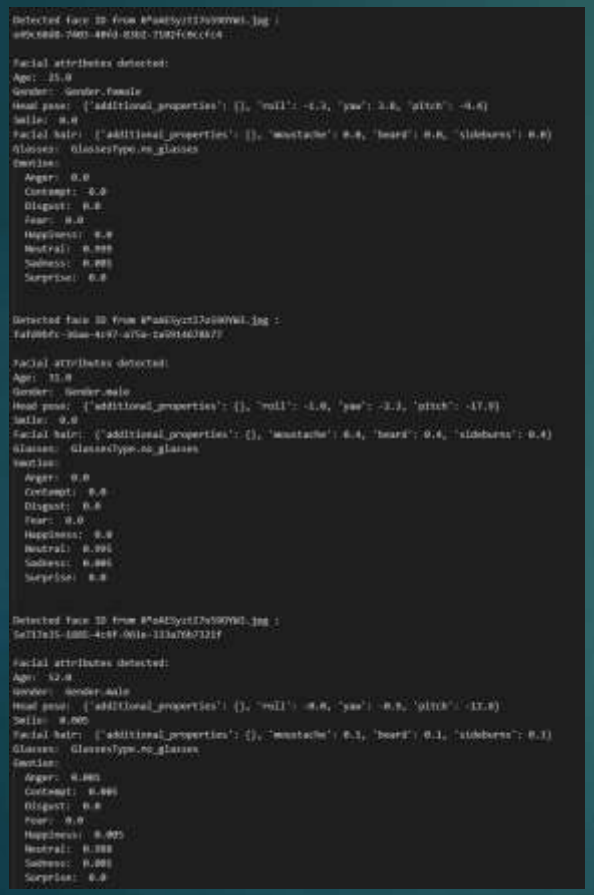


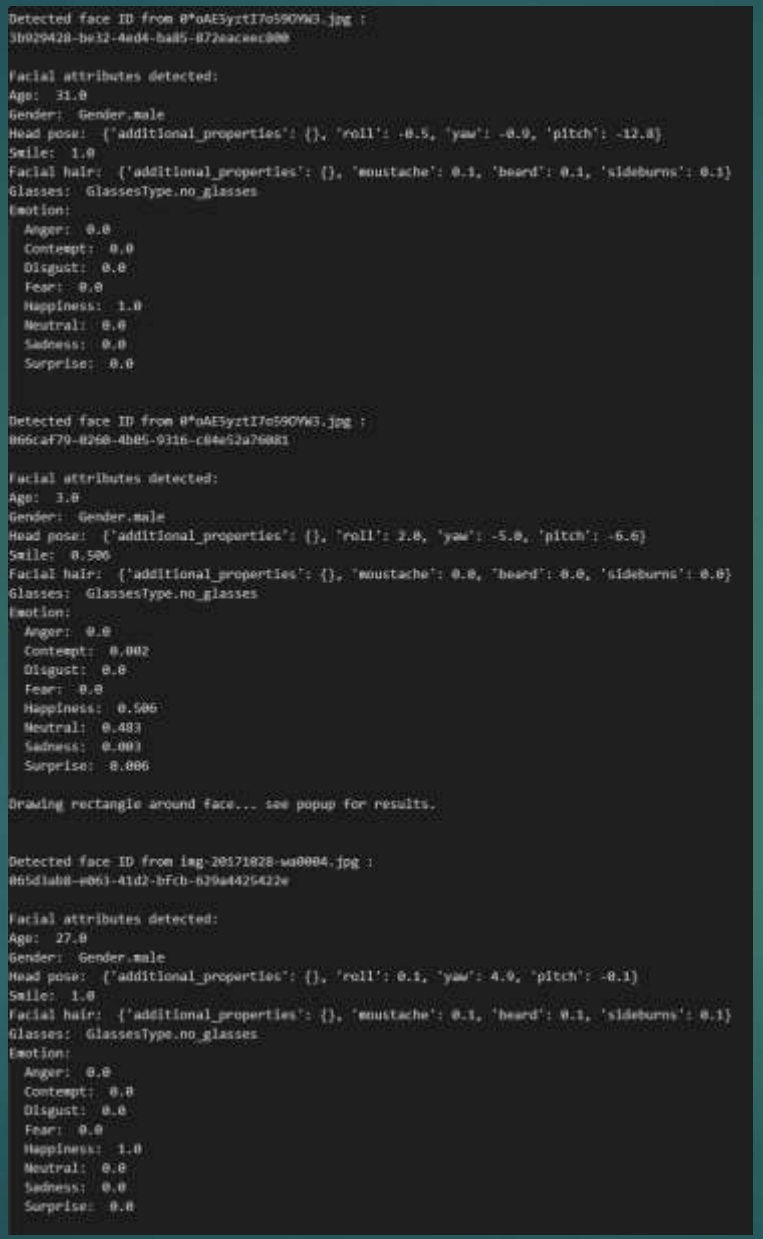
```
computervision_client = ComputerVisionClient(endpoint, CognitiveServicesCredentials(subscription_key))

'''
OCR: Read File using the Read API, extract text - remote
This example will extract text in an image, then print results, line by line.
This API call can also extract handwriting style text (not shown).
'''
print("===== Read File - remote =====")
# Get an image with text
# read_image_url = "https://media.wired.com/photos/59327d3b44db296121d6b881/master/w_1600%2Cc_limit/bond_0011_Layer-5.jpg"
read_image_url = "https://media.wired.com/photos/59327d3b44db296121d6b881/master/w_1600%2Cc_limit/bond_0011_Layer-5.jpg"
# Call API with URL and raw response (allows you to get the operation location)
read_response = computervision_client.read(read_image_url,  raw=True)

# Get the operation location (URL with an ID at the end) from the response
read_operation_location = read_response.headers["Operation-Location"]
# Grab the ID from the URL
operation_id = read_operation_location.split("/")[-1]

# Call the "GET" API and wait for it to retrieve the results
while True:
    read_result = computervision_client.get_read_result(operation_id)
    if read_result.status not in ['notStarted', 'running']:
        break
    time.sleep(1)

# Print the detected text, line by line
if read_result.status == OperationStatusCodes.succeeded:
    for text_result in read_result.analyze_result.read_results:
        for line in text_result.lines:
            print(line.text)
            print(line.bounding_box)
print()
```

```
===== Read File - remote =====
BEN MILLER
[416.0, 42.0, 582.0, 43.0, 582.0, 68.0, 416.0, 67.0]
Dearsam.
[88.0, 167.0, 292.0, 169.0, 292.0, 215.0, 88.0, 212.0]
Welcome onboard . I'm very excited to have you
[125.0, 251.0, 891.0, 249.0, 892.0, 302.0, 125.0, 304.0]
on our team for this project. your skills will
[120.0, 313.0, 818.0, 308.0, 818.0, 360.0, 120.0, 364.0]
certainly be a valued asset and I'm looking
[122.0, 370.0, 805.0, 368.0, 805.0, 415.0, 122.0, 417.0]
forward to seeing what you come up with
[113.0, 427.0, 824.0, 427.0, 824.0, 470.0, 113.0, 472.0]
Best,
[86.0, 514.0, 235.0, 514.0, 235.0, 562.0, 86.0, 563.0]
Ben
[93.0, 570.0, 242.0, 571.0, 241.0, 613.0, 93.0, 611.0]
```


tmpe10om20e.PNG
84%

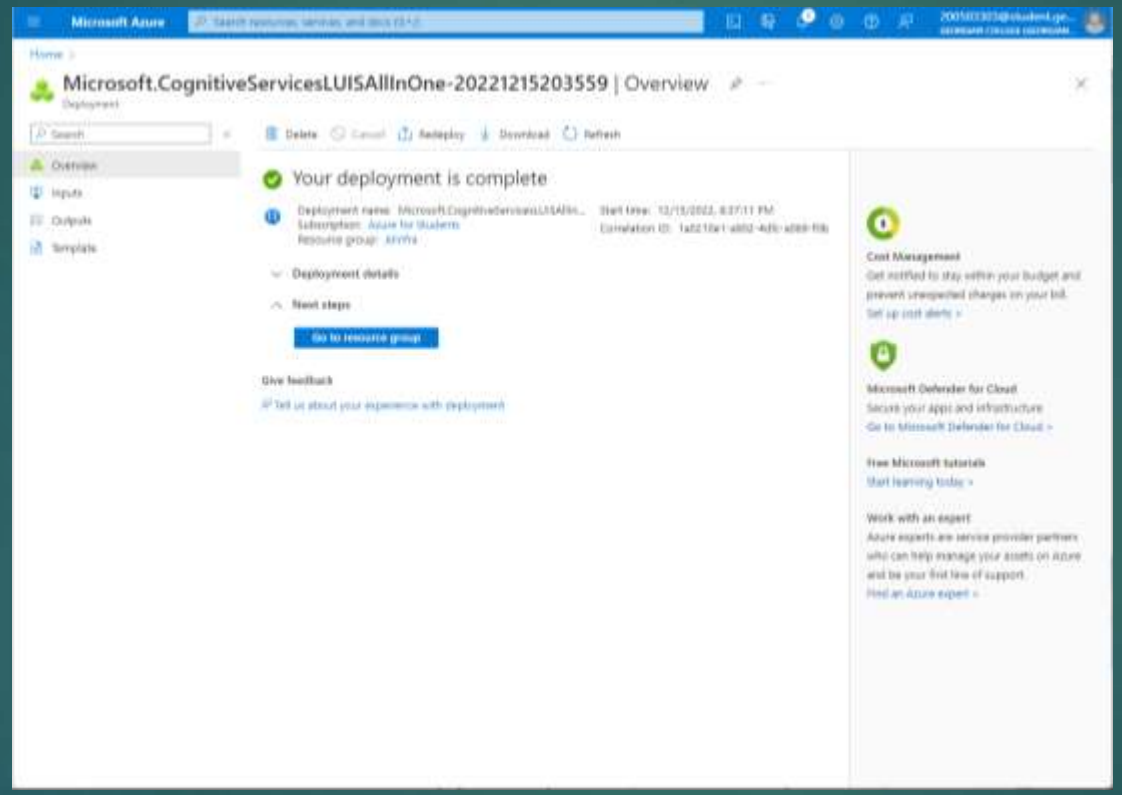

tmp8kvjnpj0.PNG
88%

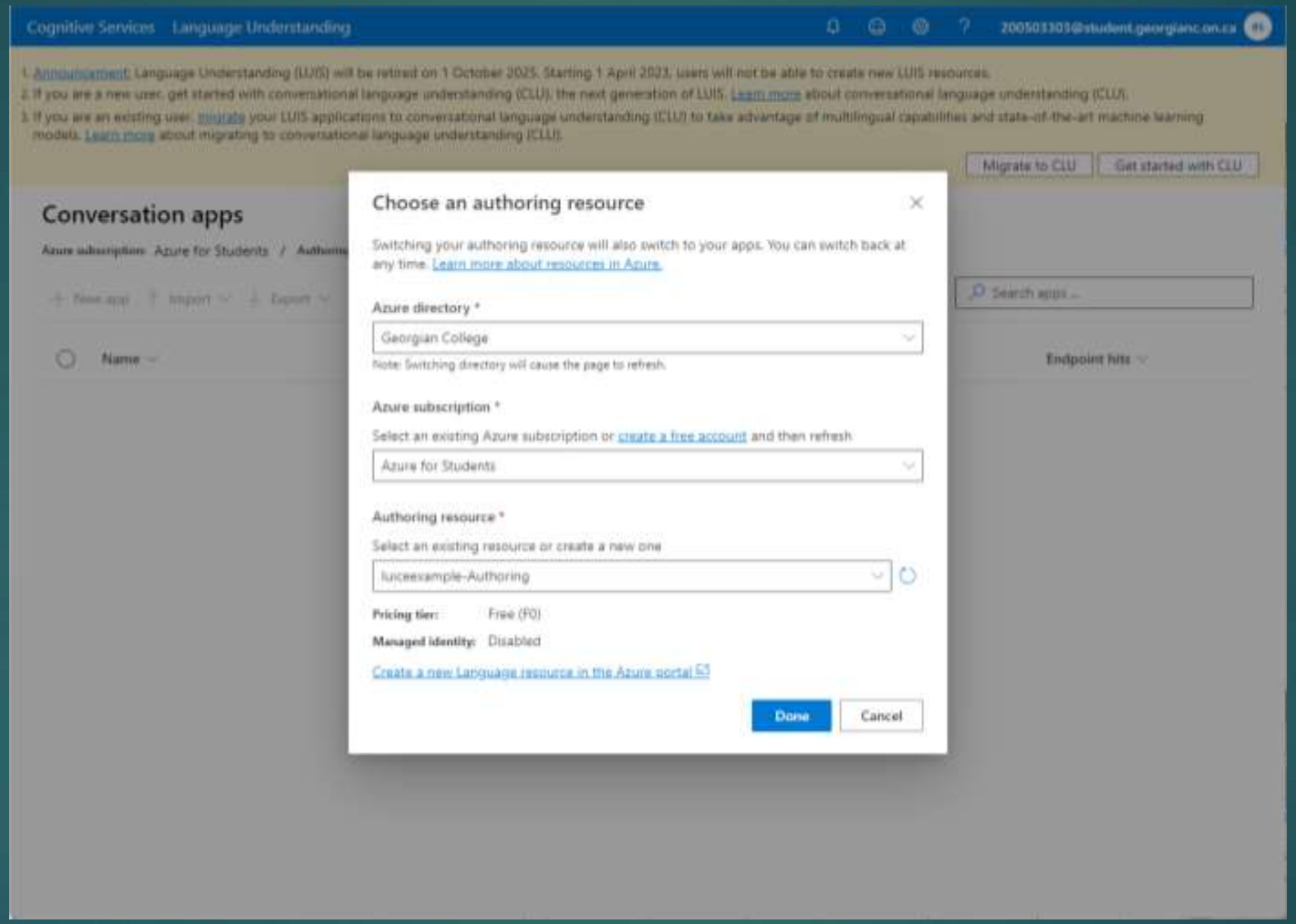


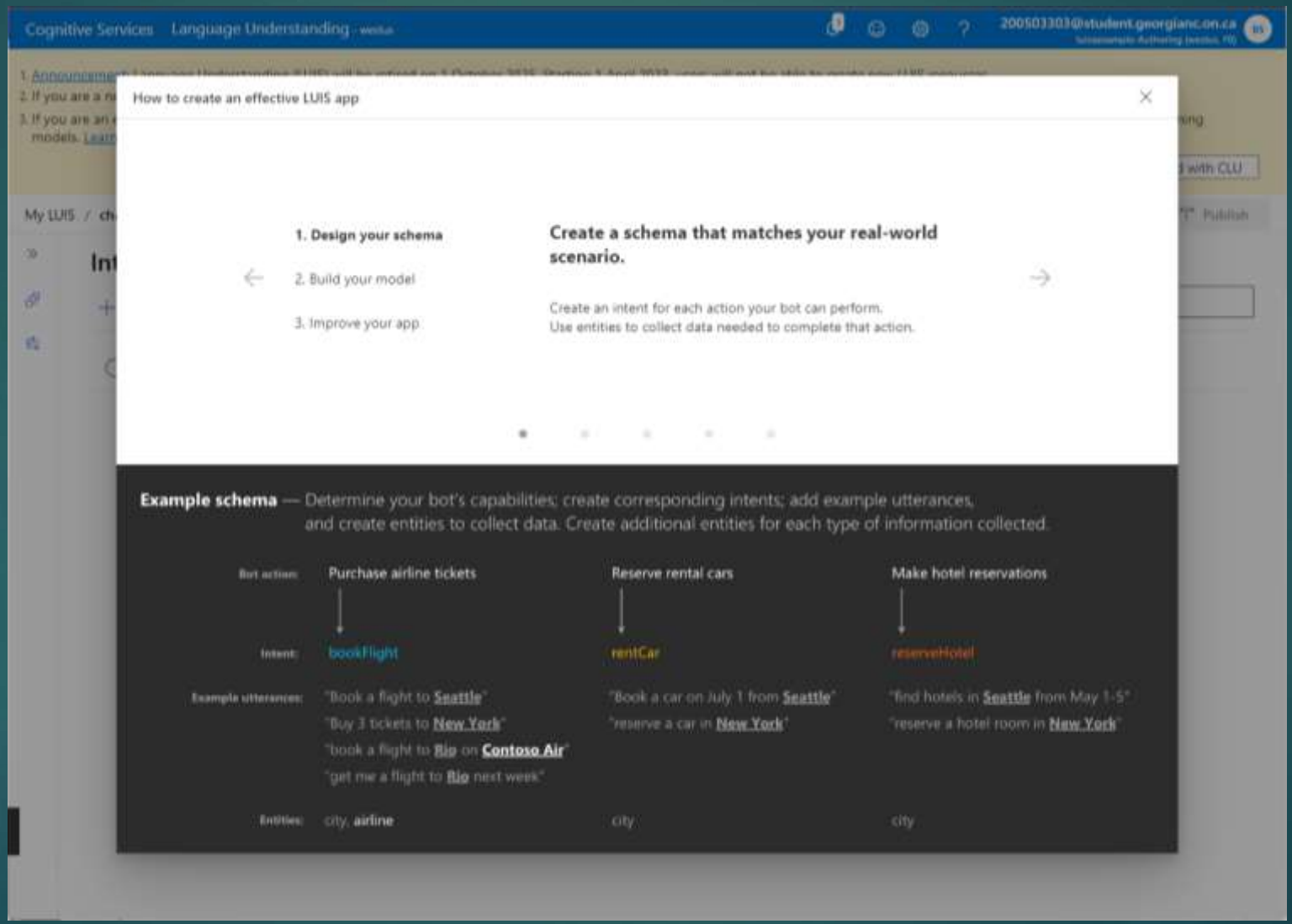


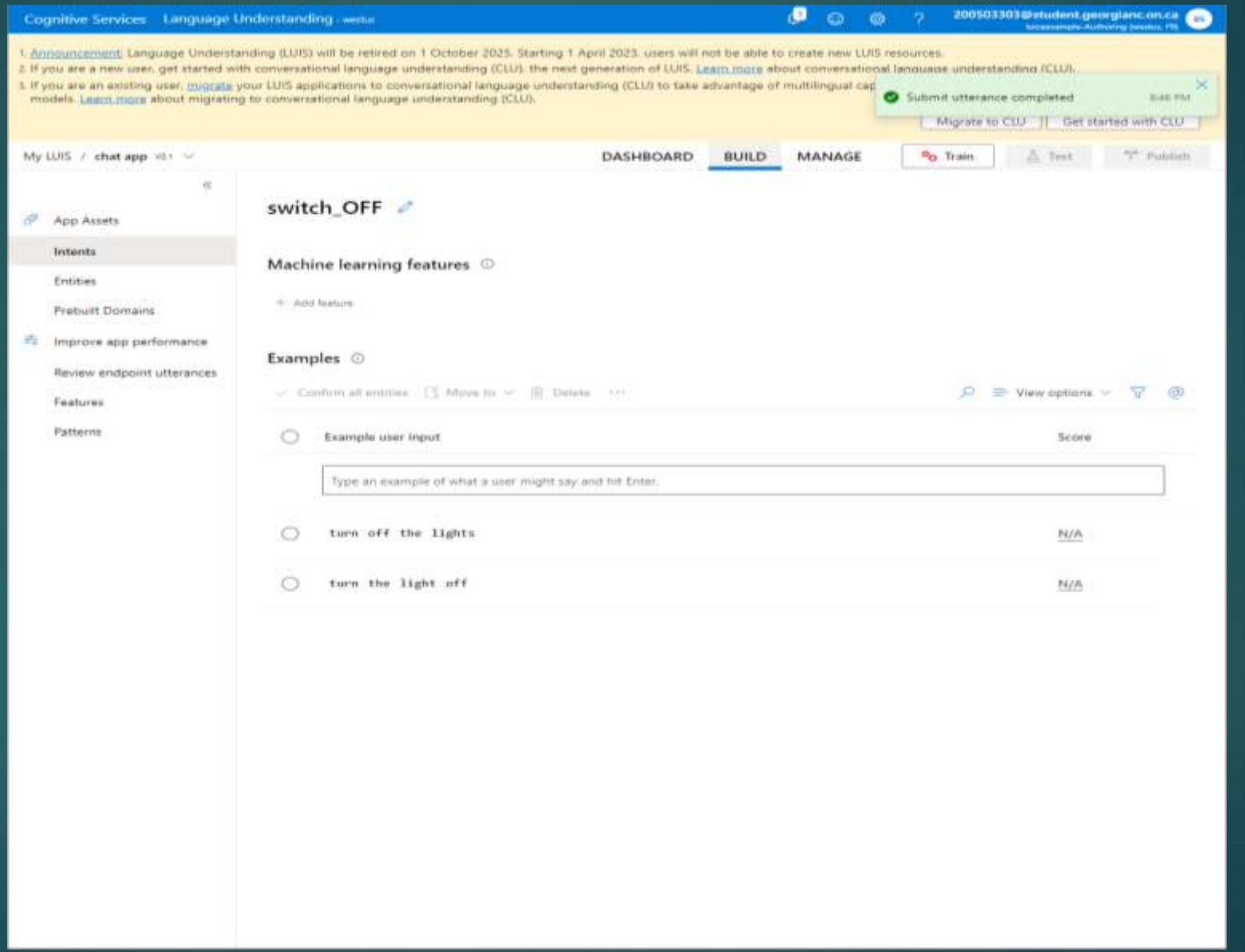

```
Detected face ID from img-20171028-wa0004.jpg :
05d57470-ebcb-449c-b0a3-e9fca4398ab2

Facial attributes detected:
Age:  29.0
Gender:  Gender.female
Head pose:  {'additional_properties': {}, 'roll': 4.9, 'yaw': -20.7, 'pitch': -0.0}
Smile:  0.992
Facial hair:  {'additional_properties': {}, 'moustache': 0.0, 'beard': 0.0, 'sideburns': 0.0}
Glasses:  GlassesType.no_glasses
Emotion:
  Anger:  0.0
  Contempt:  0.0
  Disgust:  0.0
  Fear:  0.0
  Happiness:  0.992
  Neutral:  0.008
  Sadness:  0.0
  Surprise:  0.0

Drawing rectangle around face... see popup for results.
```

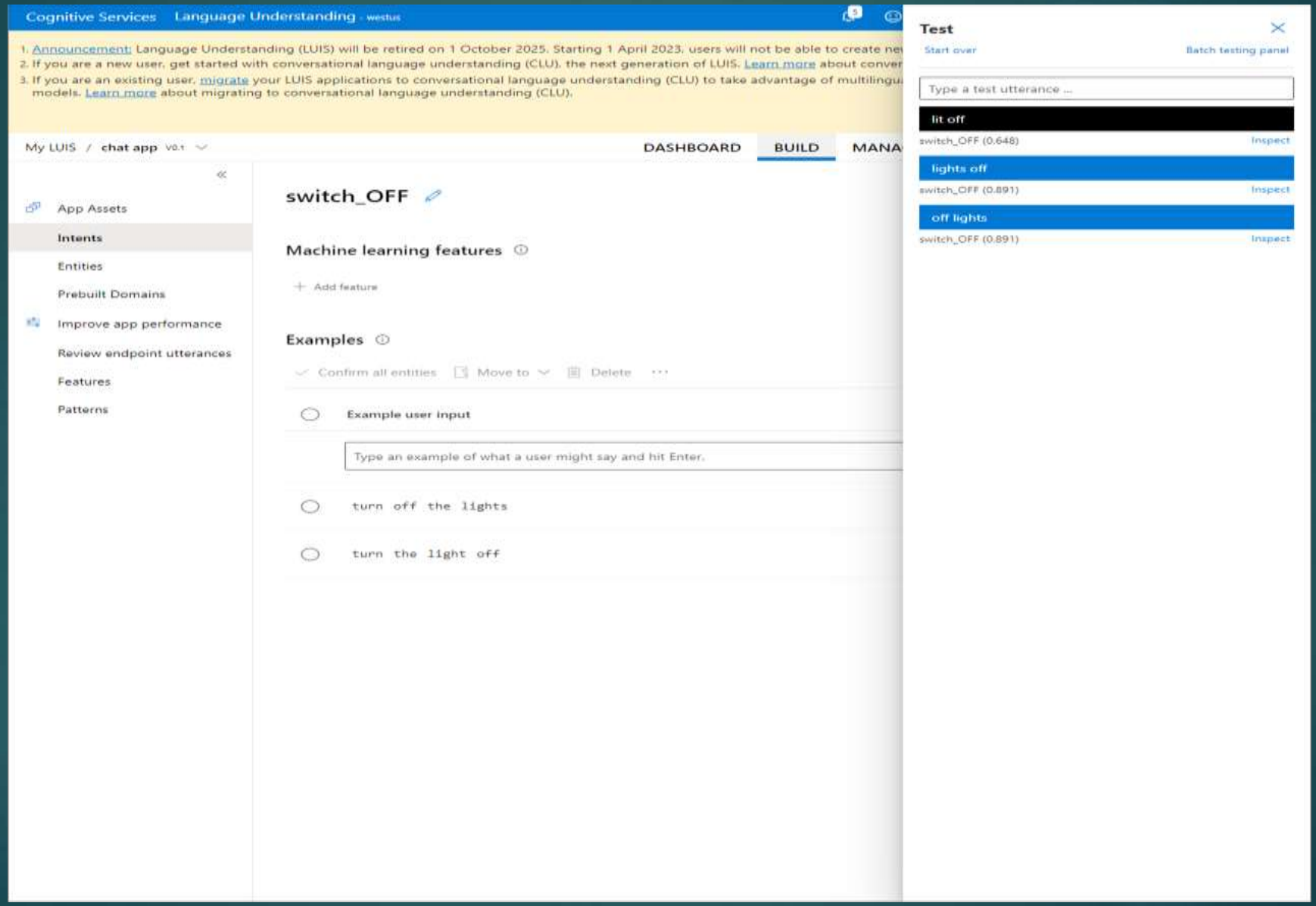
Challenge 4.10 : LUIS

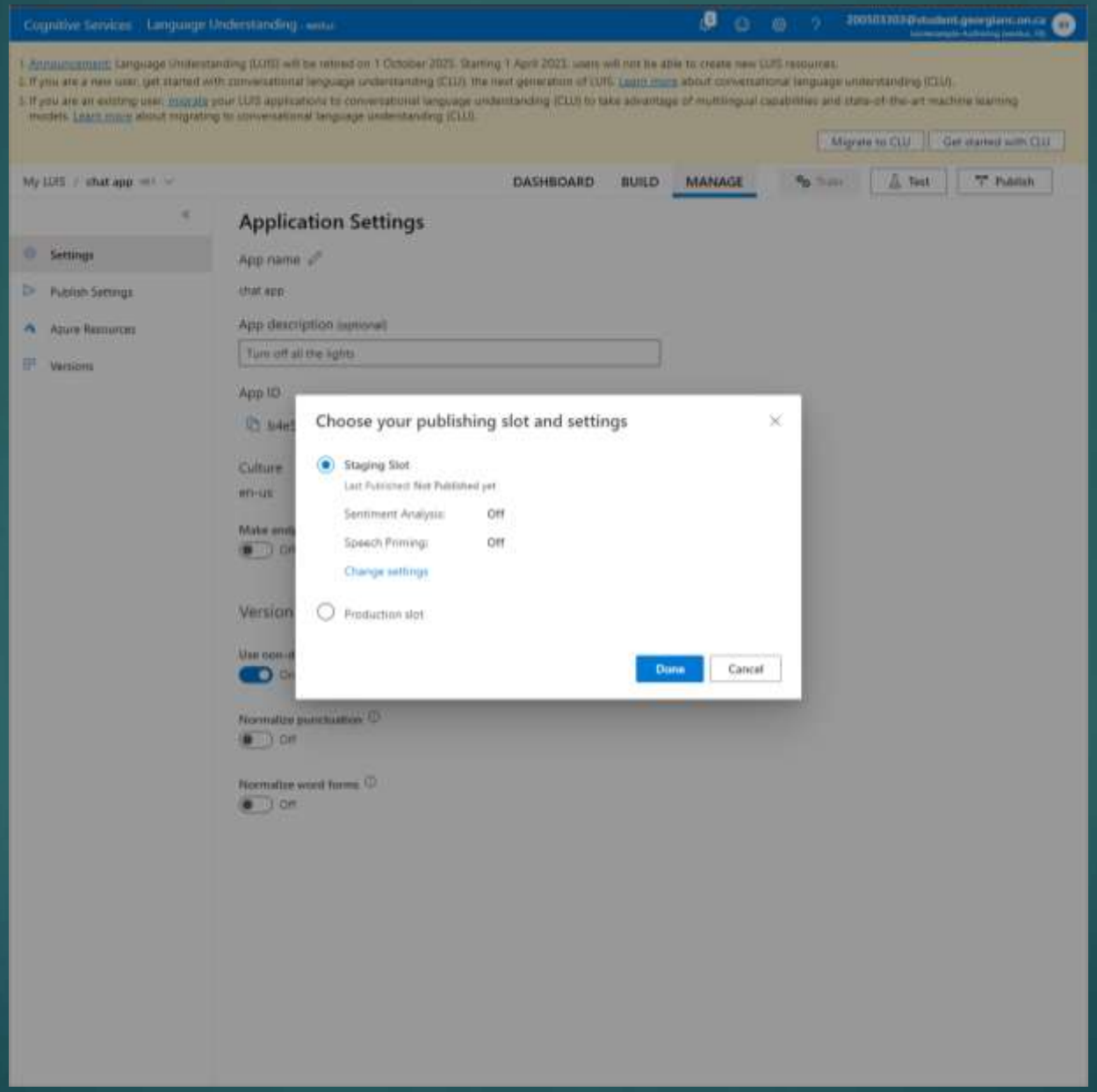


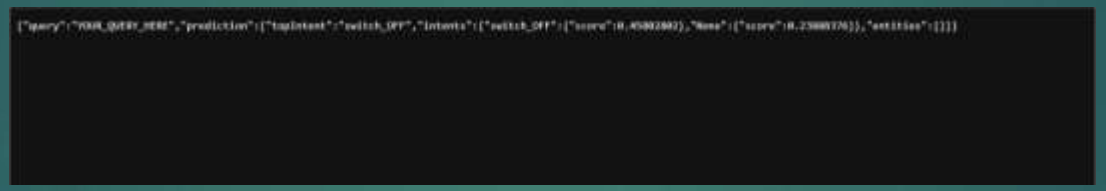

Cognitive Services | Language Understanding
Announcement: Language Understanding (LUIS) will be retired on 1 October 2025. Starting 1 April 2023, users will not be able to create new LUIS resources.
If you are a new user, get started with conversational language understanding (CLU), the next generation of LUIS. Learn more about conversational language understanding (CLU).
If you are an existing user, migrate your LUIS applications to conversational language understanding (CLU) to take advantage of multilingual capabilities and state-of-the-art machine learning models. Learn more about migrating to conversational language understanding (CLU).
Migrate to CLU
Get started with CLU
Conversation apps
Azure subscription: Azure for Students
New app
Import
Export
Search apps
Name
Endpoint hits
Choose an authoring resource
Switching your authoring resource will also switch to your apps. You can switch back at any time. Learn more about resources in Azure
Azure directory *
Georgian College
Note: Switching directory will cause the page to refresh.
Azure subscription *
Select an existing Azure subscription or create a free account and then refresh
Azure for Students
Authoring resource *
Select an existing resource or create a new one
luceexample-Authoring
Pricing tier: Free (F0)
Managed identity: Disabled
Create a new Language resource in the Azure portal
Done
Cancel

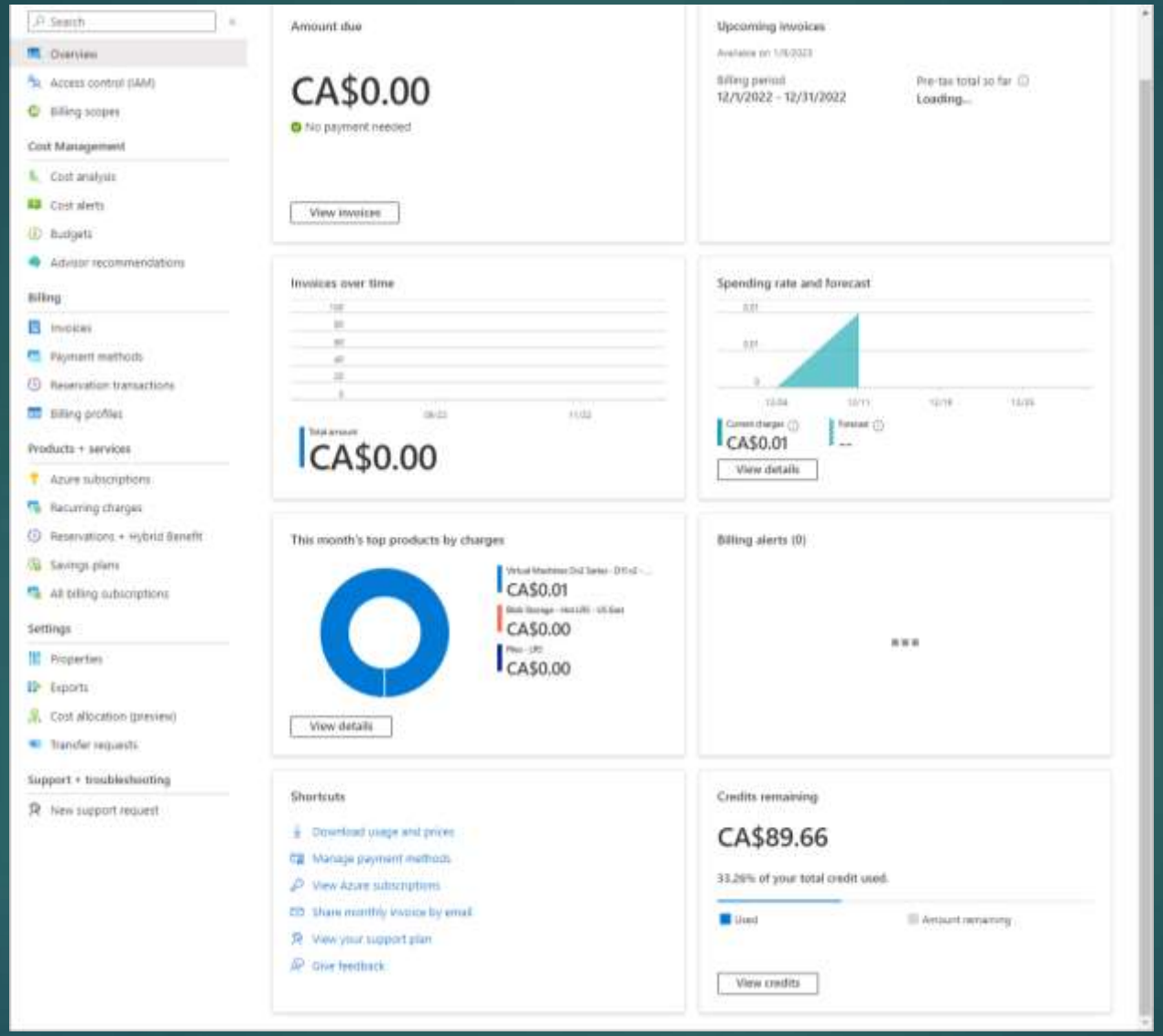

Cognitive Services
Language Understanding
How to create an effective LUIS app
1. Design your schema
2. Build your model
3. Improve your app
Create a schema that matches your real-world scenario.
Create an intent for each action your bot can perform.
Use entities to collect data needed to complete that action.
Example schema — Determine your bot's capabilities; create corresponding intents; add example utterances, and create entities to collect data. Create additional entities for each type of information collected.
Bot action:
Purchase airline tickets
Reserve rental cars
Make hotel reservations
Intent:
bookFlight
rentCar
reserveHotel
Example utterances:
"Book a flight to Seattle"
"Buy 3 tickets to New York"
"book a flight to Rio on Contoso Air"
"get me a flight to Rio next week"
"Book a car on July 1 from Seattle"
"reserve a car in New York"
"find hotels in Seattle from May 1-5"
"reserve a hotel room in New York"
Entities:
city, airline
city
city


Cognitive Services Language Understanding
Announcement: Language Understanding (LUIS) will be retired on 1 October 2025. Starting 1 April 2023, users will not be able to create new LUIS resources.
If you are a new user, get started with conversational language understanding (CLU), the next generation of LUIS. Learn more about conversational language understanding (CLU).
If you are an existing user, migrate your LUIS applications to conversational language understanding (CLU) to take advantage of multilingual cap
models. Learn more about migrating to conversational language understanding (CLU).
Submit utterance completed
Migrate to CLU
Get started with CLU
My LUIS / chat app
DASHBOARD
BUILD
MANAGE
Train
Test
Publish
App Assets
Intents
Entities
Prebuilt Domains
Improve app performance
Review endpoint utterances
Features
Patterns
switch_OFF
Machine learning features
Add feature
Examples
Confirm all entities
Move to
Delete
View options
Example user input
Score
Type an example of what a user might say and hit Enter.
turn off the lights
N/A
turn the light off
N/A


Cognitive Services
Language Understanding
westus
1. Announcement: Language Understanding (LUIS) will be retired on 1 October 2025. Starting 1 April 2023, users will not be able to create ne
2. If you are a new user, get started with conversational language understanding (CLU), the next generation of LUIS. Learn more about conver
3. If you are an existing user, migrate your LUIS applications to conversational language understanding (CLU) to take advantage of multilingu
models. Learn more about migrating to conversational language understanding (CLU).
My LUIS / chat app v0.1
DASHBOARD
BUILD
MANA
App Assets
Intents
Entities
Prebuilt Domains
Improve app performance
Review endpoint utterances
Features
Patterns
switch_OFF
Machine learning features
Add feature
Examples
Confirm all entities
Move to
Delete
Example user input
Type an example of what a user might say and hit Enter.
turn off the lights
turn the light off
Test
Start over
Batch testing panel
Type a test utterance ...
lit off
switch_OFF (0.648)
Inspect
lights off
switch_OFF (0.891)
Inspect
off lights
switch_OFF (0.891)
Inspect


Cognitive Services | Language Understanding
DASHBOARD
BUILD
MANAGE
Test
Publish
Application Settings
App name
chat app
App description (optional)
Turn off all the lights
App ID
Culture
en-us
Settings
Publish Settings
Azure Resources
Versions
Choose your publishing slot and settings
Staging Slot
Last Published: Not Published yet
Sentiment Analysis: Off
Speech Priming: Off
Change settings
Production slot
Done
Cancel
Normalize punctuation
Off
Normalize word forms
Off


{"query":"YOUR_QUERY_HERE","prediction":{"topIntent":"switch_OFF","intents":{"switch_OFF":{"score":0.45002002},"None":{"score":0.23008376}},"entities":{}}}


Search
Overview
Access control (IAM)
Billing scopes
Cost Management
Cost analysis
Cost alerts
Budgets
Advisor recommendations
Billing
Invoices
Payment methods
Reservation transactions
Billing profiles
Products + services
Azure subscriptions
Recurring charges
Reservations + Hybrid Benefit
Savings plans
All billing subscriptions
Settings
Properties
Exports
Cost allocation (preview)
Transfer requests
Support + troubleshooting
New support request
Amount due
CA$0.00
No payment needed
View invoices
Upcoming invoices
Billing period
12/1/2022 - 12/31/2022
Pre-tax total so far
Loading...
Invoices over time
CA$0.00
Spending rate and forecast
CA$0.01
View details
This month's top products by charges
CA$0.01
CA$0.00
CA$0.00
View details
Billing alerts (0)
Shortcuts
Download usage and prices
Manage payment methods
View Azure subscriptions
Share monthly invoice by email
View your support plan
Give feedback
Credits remaining
CA$89.66
33.26% of your total credit used.
Used
Amount remaining
View credits

Thank YOU