Final Project

Face API

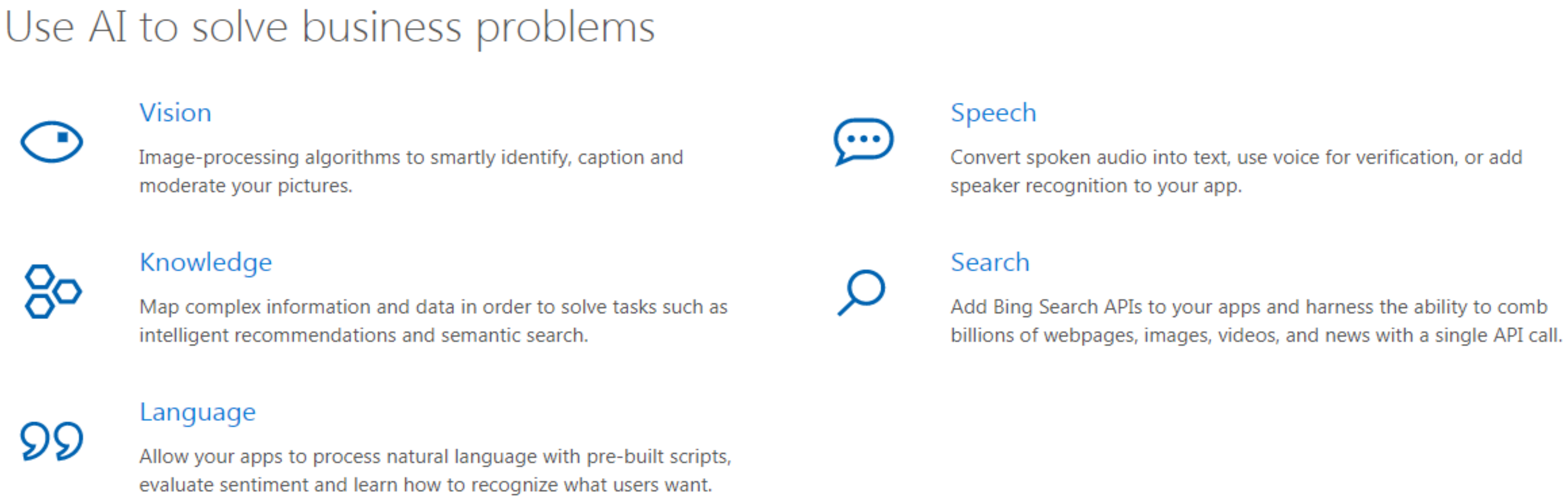
Patil, Aniruddha

Deep Azure@McKesson
Dr. Zoran B. Djordjević

Introduction

- **Topic:** Demonstration of Azure Face API

- **Problem:** To collect data intelligence from people's faces , their expressions and kind of attendance for a specific public gathering/event with use of Azure Cognitive Services - Face API (Cognitive Services- Vision), Azure Storage and Visualization with Pandas.



Use AI to solve business problems

Vision

Image-processing algorithms to smartly identify, caption and moderate your pictures.

Speech

Convert spoken audio into text, use voice for verification, or add speaker recognition to your app.

Knowledge

Map complex information and data in order to solve tasks such as intelligent recommendations and semantic search.

Search

Add Bing Search APIs to your apps and harness the ability to comb billions of webpages, images, videos, and news with a single API call.

Language

Allow your apps to process natural language with pre-built scripts, evaluate sentiment and learn how to recognize what users want.

Introduction

- **Technology:**

  Azure Cognitive Services, Azure Storage- Tables, Pandas

- **Hardware:** Intel i5-6300U CPU 2.4Ghz, 16 GB RAM, 64 bit Windows 7 OS

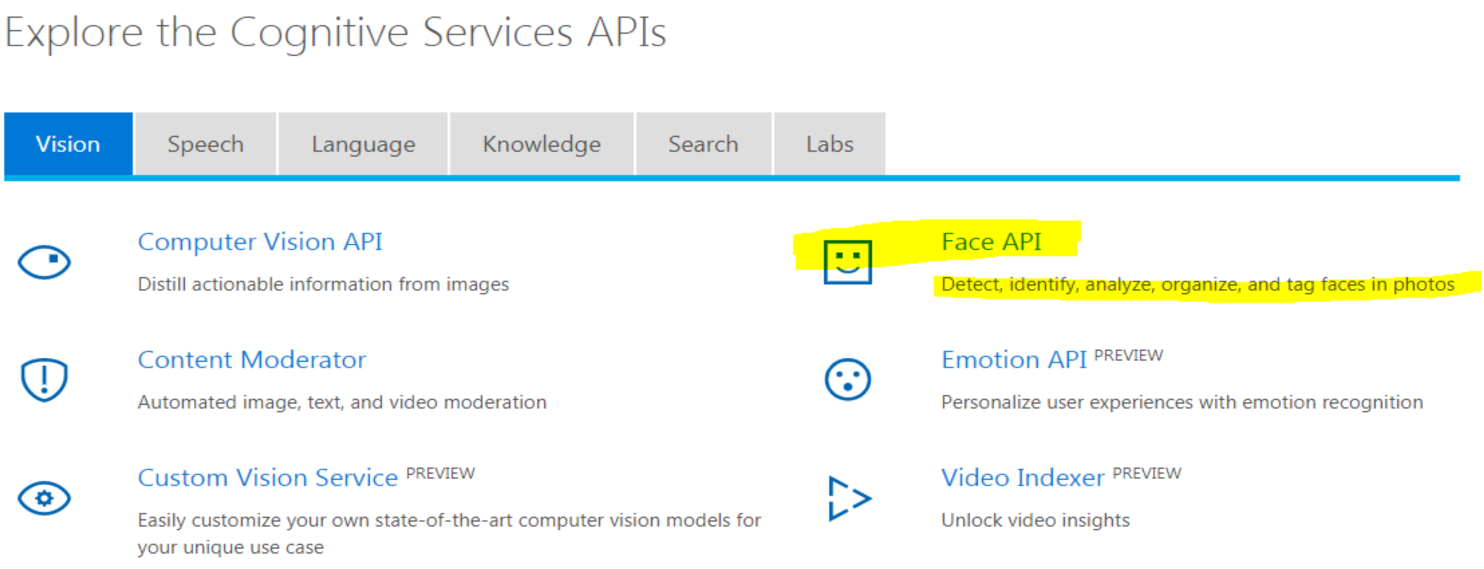
- **Software:**

Technology / Tools	Description
Azure Cognitive Services – Face API	Azure's Face API services
Azure Storage - Table	Azure's Storage services - Table
Python 2.7 & 3.6	Python 2.7 & 3.6 (Used 2.7 for Visualization due to 3.6 installation issues)
Microsoft Azure Storage Explorer	To manage Azure Table data

Azure Cognitive Services – Face API

- **Narrative:**

Microsoft Face API is a cloud-based service that provides the most advanced face algorithms. Face API has two main functions: face detection with attributes and face recognition.



Explore the Cognitive Services APIs

Computer Vision API

Distill actionable information from images

Face API

Detect, identify, analyze, organize, and tag faces in photos

Content Moderator

Automated image, text, and video moderation

Emotion API PREVIEW

Personalize user experiences with emotion recognition

Custom Vision Service PREVIEW

Easily customize your own state-of-the-art computer vision models for your unique use case

Video Indexer PREVIEW

Unlock video insights

Azure Face API – Pros, Cons and Lessons Learnt

- **Pros :**

- Azure Face API accurately delivers face attributes and specially age

- Face API can detect very small faces in images correctly. In one of my test image, it captured a face in picture on shirt person wearing in image

- Free pricing Tier is available for use

- **Cons:**

- Azure Face API does not work with RAW image formats produced by cameras. It only works with JPG, BMP, GIF and PNG

- Free tier has limitation to run 20 calls in a minute

- **Lessons Learnt :**

- There are multiple image data sources available to test with but lot of images are in non JPG format mostly RAW format. I could not get Python packages working with non JPG format images. I also saw that Azure Face API only works with JPG, BMP, PNG and GIF files.

- Azure Face API (Free pricing Tier) can only run 20 calls per minute. So when I was running for big batch, I was getting errors. I reduced down my input to < 20 for Demo. We can use time delay function to overcome this limit to some extent.

Azure Cognitive Services – Face API

- **Narrative:** (continued)

Face Detection

Face API detects human faces with high precision face location in an image. Face rectangle indicating the face location in the image is returned along with each detected face and a series of face related attributes such as pose, gender, age, head pose, facial hair and glasses.

Face Recognition

Face recognition is widely used in many scenarios including security, natural user interface, image content analysis and management, mobile apps, and robotics. Four face recognition functions are provided: face verification, finding similar faces, face grouping, and person identification.

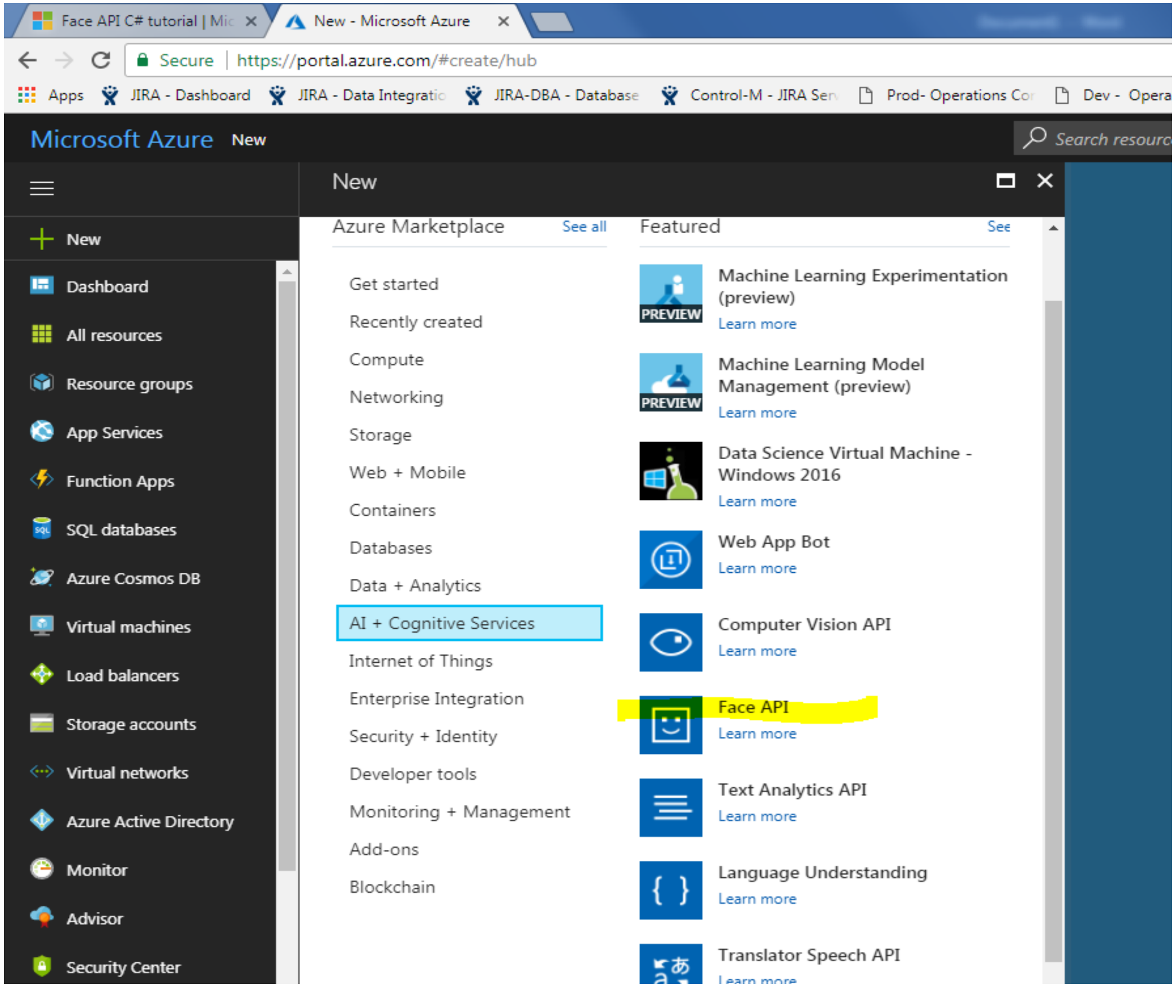
Azure Cognitive Services – Face API

- **Face Detection Demonstration Steps:**

- 1. Subscribe to Face API
- 2. Download Image Database for testing Face API
- 3. Setup Azure Storage Account to be used for storing all image attributes
- 4. Learn Face API.
- 5. Code development to use Face API for Face Detection and collect face attributes in Azure Storage
- 6. Data Visualization & inference

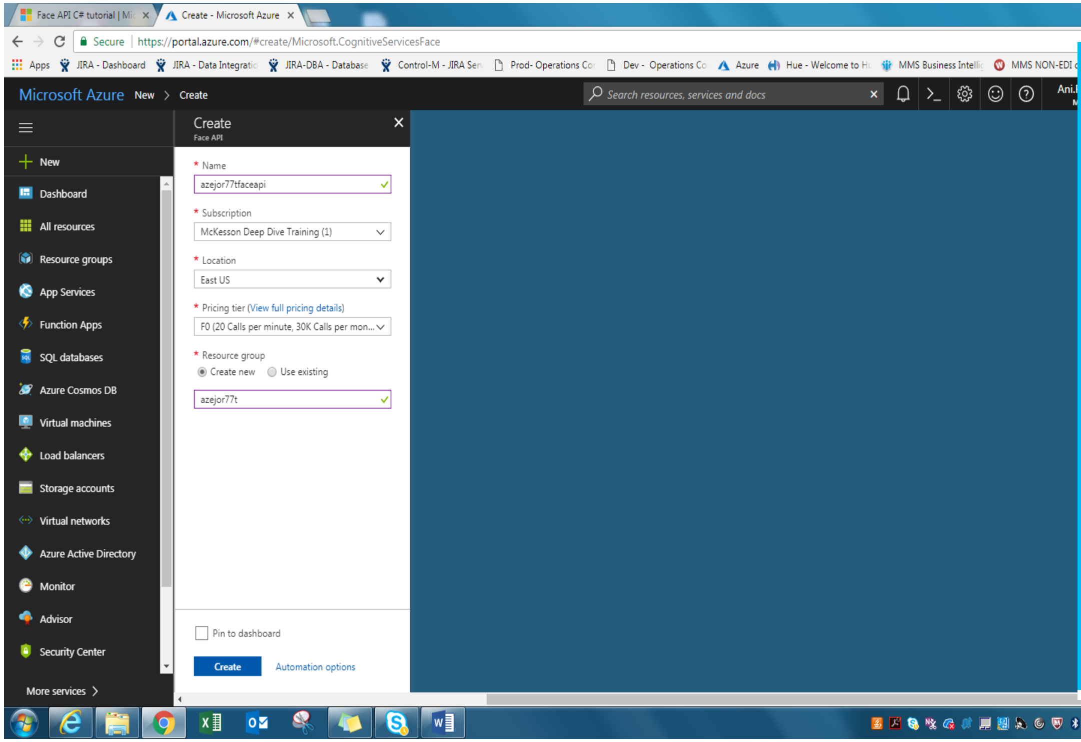
Subscribe to Face API

- Login to Azure Portal and subscribe to Face API



Subscribe to Face API

- Below see that we have selected Free pricing Ties with 20 calls per minute limit



Create
Face API

* Name
azejor77tfaceapi ✓

* Subscription
McKesson Deep Dive Training (1)

* Location
East US

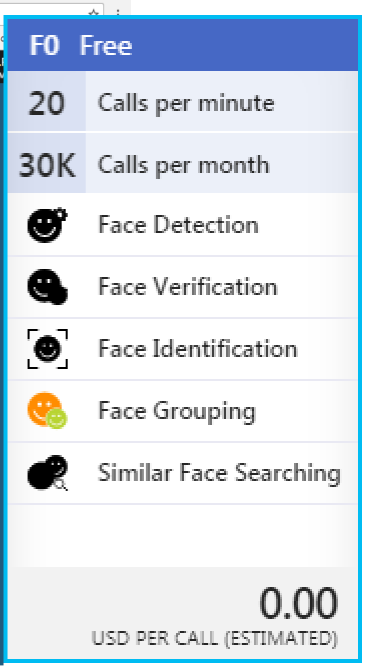
* Pricing tier (View full pricing details)
F0 (20 Calls per minute, 30K Calls per mon...

* Resource group
● Create new ○ Use existing

azejor77t ✓

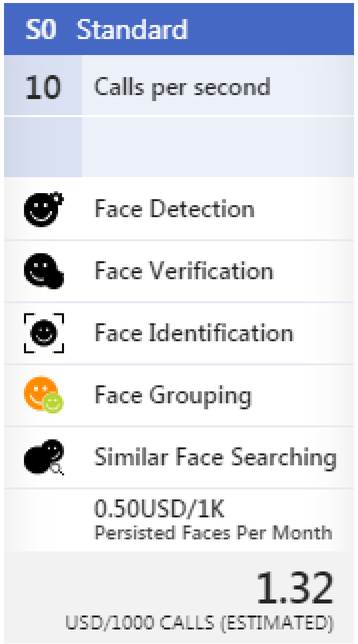
□ Pin to dashboard

Create   Automation options



F0	Free
20	Calls per minute
30K	Calls per month
	Face Detection
	Face Verification
	Face Identification
	Face Grouping
	Similar Face Searching

0.00
USD PER CALL (ESTIMATED)

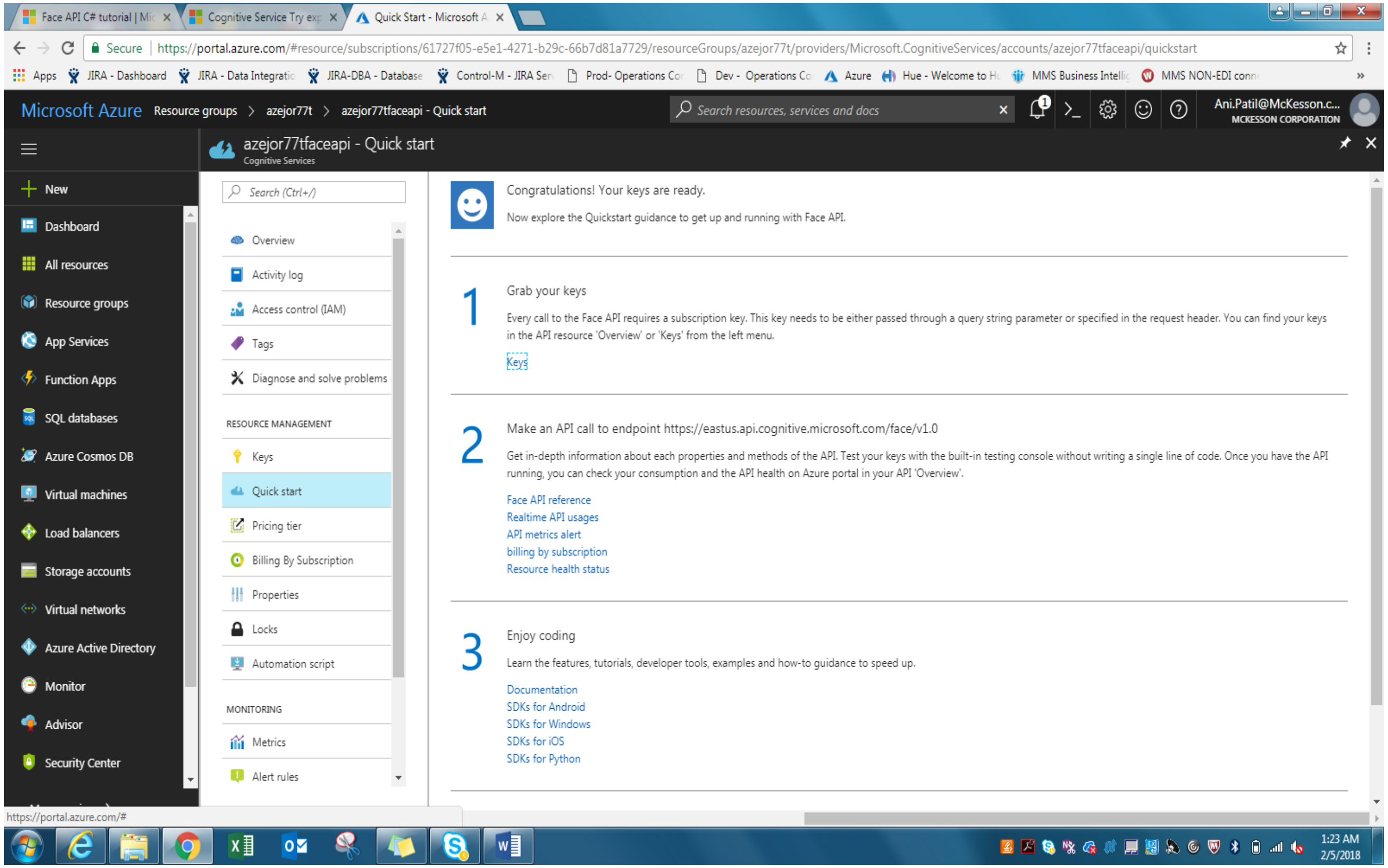


S0	Standard
10	Calls per second
	Face Detection
	Face Verification
	Face Identification
	Face Grouping
	Similar Face Searching
	0.50USD/1K Persisted Faces Per Month

1.32
USD/1000 CALLS (ESTIMATED)

Face API

- Grab your Keys from Option 1 pointed below



Congratulations! Your keys are ready.

Now explore the Quickstart guidance to get up and running with Face API.

1 Grab your keys

Every call to the Face API requires a subscription key. This key needs to be either passed through a query string parameter or specified in the request header. You can find your keys in the API resource 'Overview' or 'Keys' from the left menu.

Keys

2 Make an API call to endpoint https://eastus.api.cognitive.microsoft.com/face/v1.0

Get in-depth information about each properties and methods of the API. Test your keys with the built-in testing console without writing a single line of code. Once you have the API running, you can check your consumption and the API health on Azure portal in your API 'Overview'.

Face API reference
Realtime API usages
API metrics alert
billing by subscription
Resource health status

3 Enjoy coding

Learn the features, tutorials, developer tools, examples and how-to guidance to speed up.

Documentation
SDKs for Android
SDKs for Windows
SDKs for iOS
SDKs for Python

Input Data Source – Image DB

- Image database

http://www.anefian.com/research/face_reco.htm

- To download, click 'Georgia Tech face database' below

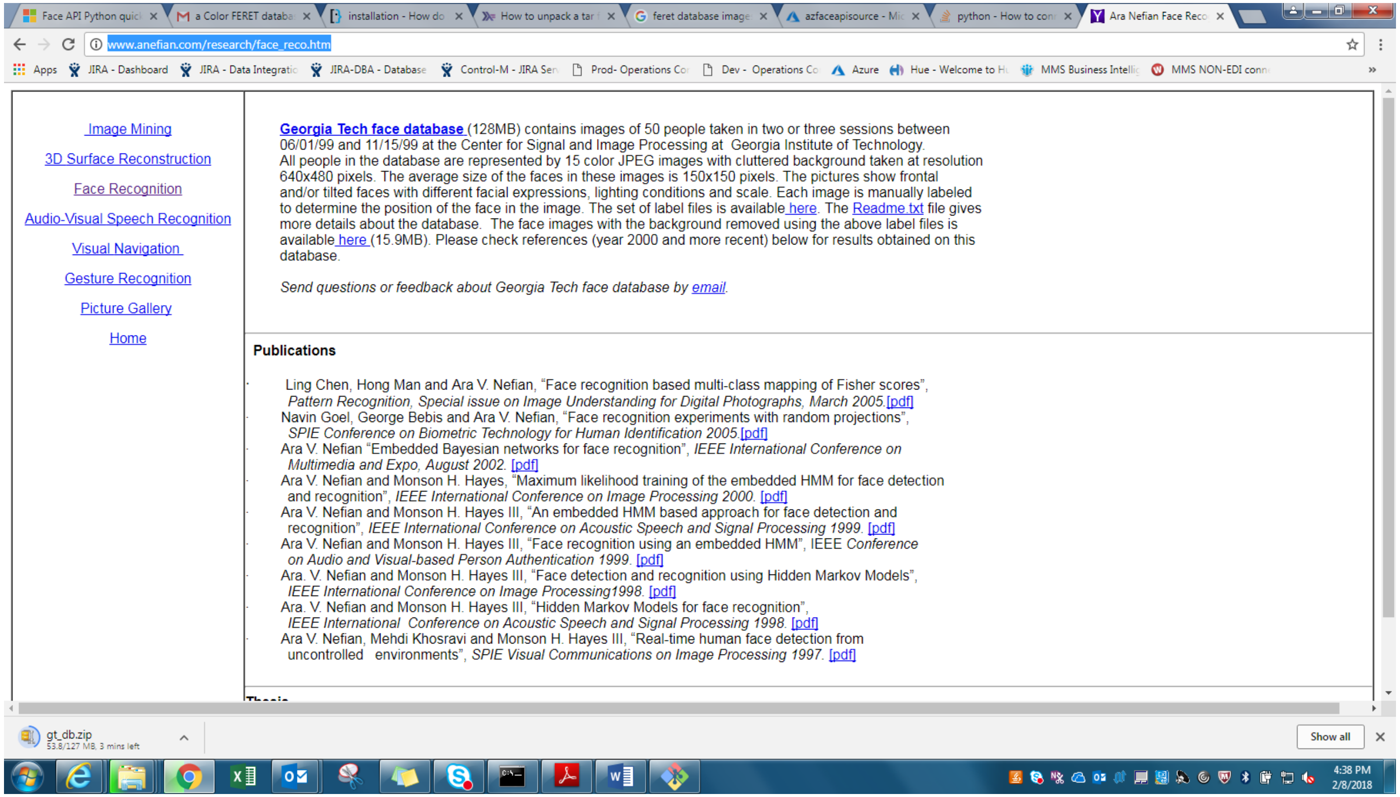


Image Mining

3D Surface Reconstruction

Face Recognition

Audio-Visual Speech Recognition

Visual Navigation

Gesture Recognition

Picture Gallery

Home

Georgia Tech face database (128MB) contains images of 50 people taken in two or three sessions between 06/01/99 and 11/15/99 at the Center for Signal and Image Processing at Georgia Institute of Technology. All people in the database are represented by 15 color JPEG images with cluttered background taken at resolution 640x480 pixels. The average size of the faces in these images is 150x150 pixels. The pictures show frontal and/or tilted faces with different facial expressions, lighting conditions and scale. Each image is manually labeled to determine the position of the face in the image. The set of label files is available here. The Readme.txt file gives more details about the database. The face images with the background removed using the above label files is available here (15.9MB). Please check references (year 2000 and more recent) below for results obtained on this database.

Send questions or feedback about Georgia Tech face database by email.

Publications

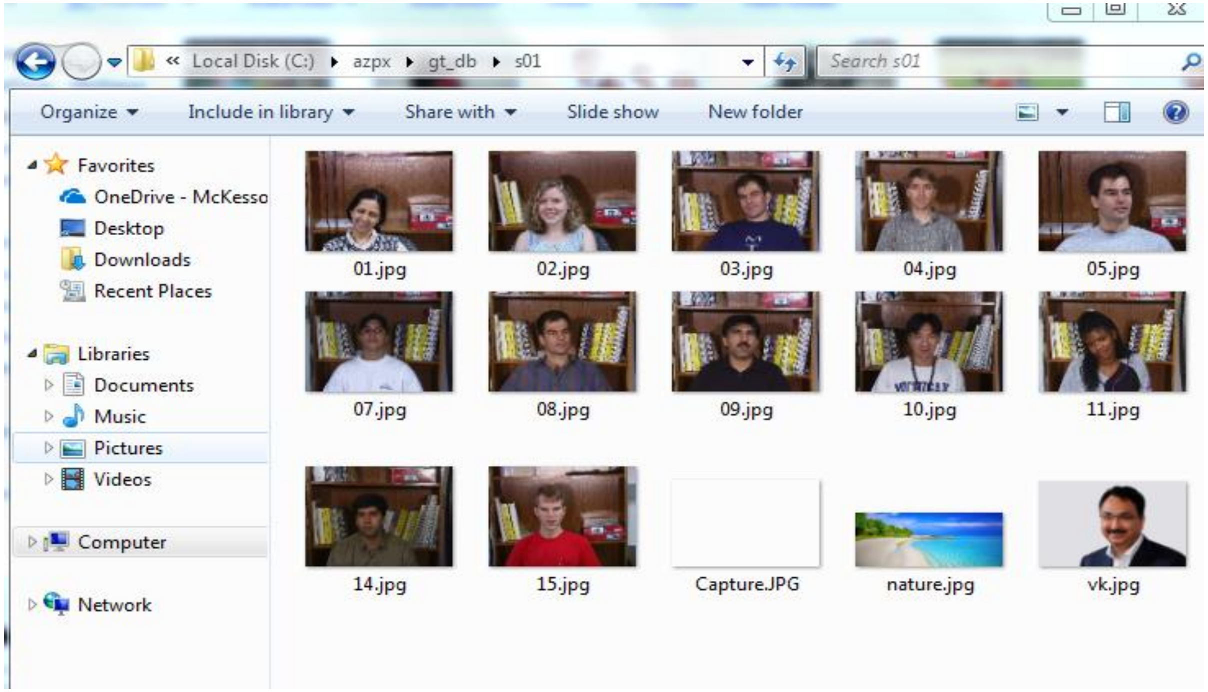
- Ling Chen, Hong Man and Ara V. Nefian, "Face recognition based multi-class mapping of Fisher scores", *Pattern Recognition, Special issue on Image Understanding for Digital Photographs, March 2005* [pdf]
- Navin Goel, George Bebis and Ara V. Nefian, "Face recognition experiments with random projections", *SPIE Conference on Biometric Technology for Human Identification 2005* [pdf]
- Ara V. Nefian "Embedded Bayesian networks for face recognition", *IEEE International Conference on Multimedia and Expo, August 2002.* [pdf]
- Ara V. Nefian and Monson H. Hayes, "Maximum likelihood training of the embedded HMM for face detection and recognition", *IEEE International Conference on Image Processing 2000.* [pdf]
- Ara V. Nefian and Monson H. Hayes III, "An embedded HMM based approach for face detection and recognition", *IEEE International Conference on Acoustic Speech and Signal Processing 1999.* [pdf]
- Ara V. Nefian and Monson H. Hayes III, "Face recognition using an embedded HMM", IEEE *Conference on Audio and Visual-based Person Authentication 1999.* [pdf]
- Ara. V. Nefian and Monson H. Hayes III, "Face detection and recognition using Hidden Markov Models", *IEEE International Conference on Image Processing1998.* [pdf]
- Ara. V. Nefian and Monson H. Hayes III, "Hidden Markov Models for face recognition", *IEEE International Conference on Acoustic Speech and Signal Processing 1998.* [pdf]
- Ara V. Nefian, Mehdi Khosravi and Monson H. Hayes III, "Real-time human face detection from uncontrolled environments", *SPIE Visual Communications on Image Processing 1997.* [pdf]

Thesis

gt_db.zip
53.8/127 MB, 3 mins left

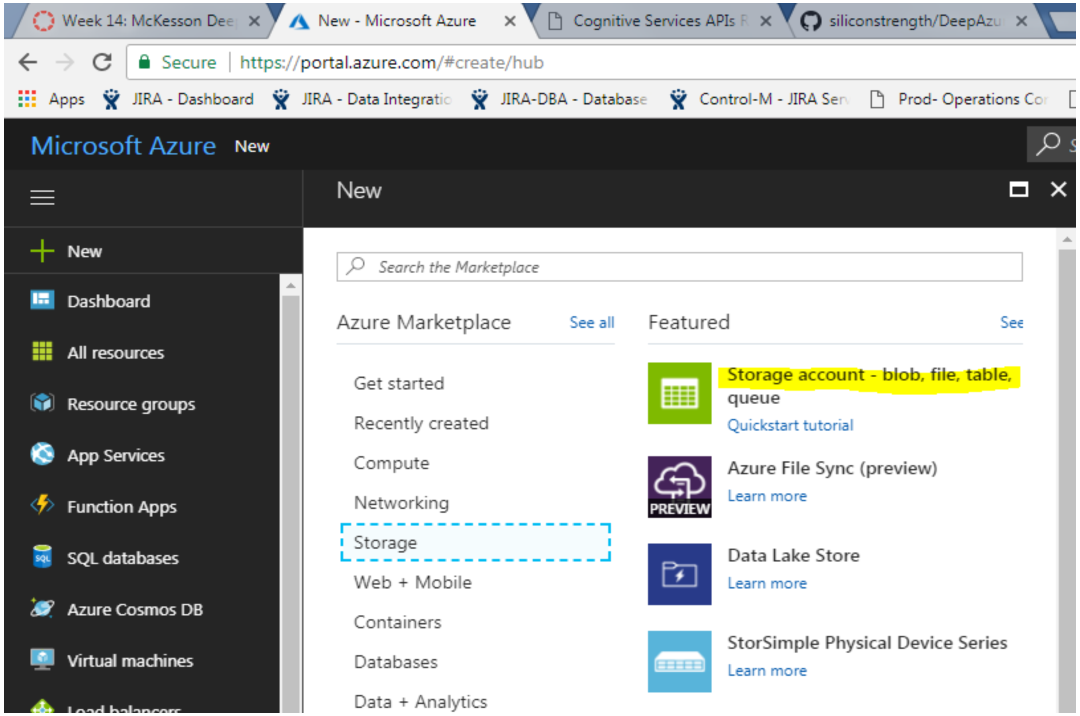
Input Data Source – Image DB

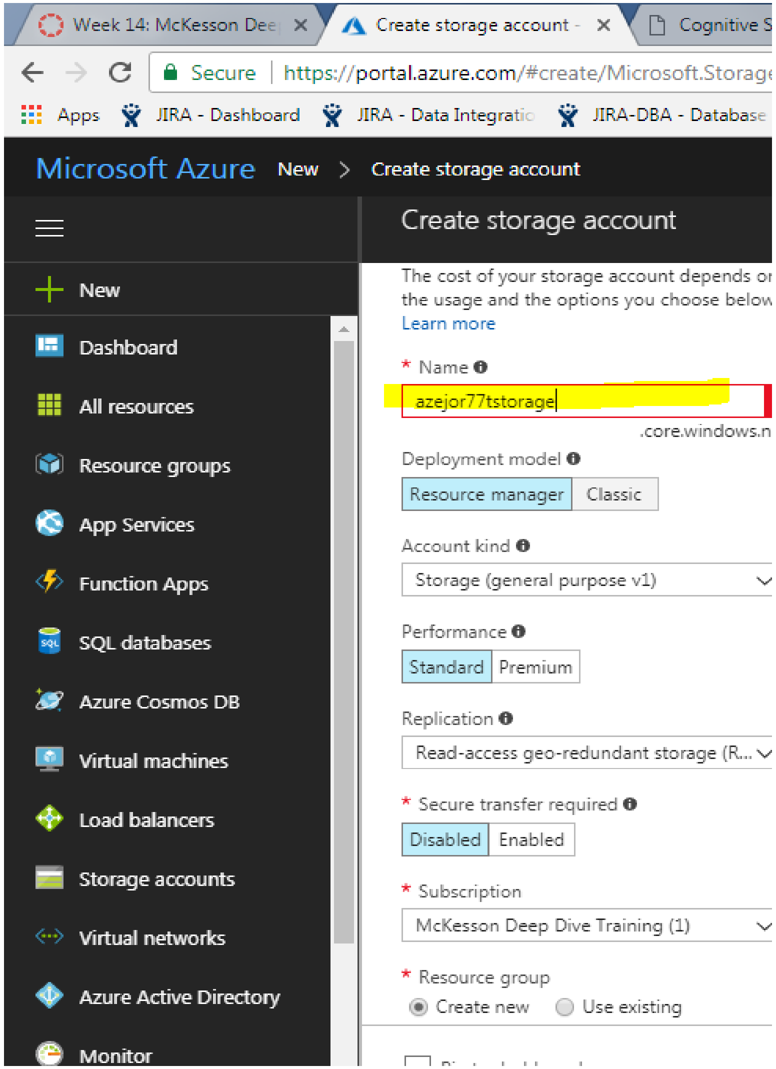
- This Database has JPG images of multiple people in different lighting conditions. Complete database is about 128MB in size.
- Download image database and take subset of images in a directory for Project demo. Below is a subset I used for this demo.
- If you see below input images, I have added some random pictures which does not have any face in them. This is to test if Face API identify it correctly to not detect faces.



Create Azure Storage Account

- Create Storage Account to store Face attribute results.





Create storage account

The cost of your storage account depends or the usage and the options you choose below Learn more

* Name: azejor77tstorage .core.windows.n

Deployment model: Resource manager | Classic

Account kind: Storage (general purpose v1)

Performance: Standard | Premium

Replication: Read-access geo-redundant storage (R...

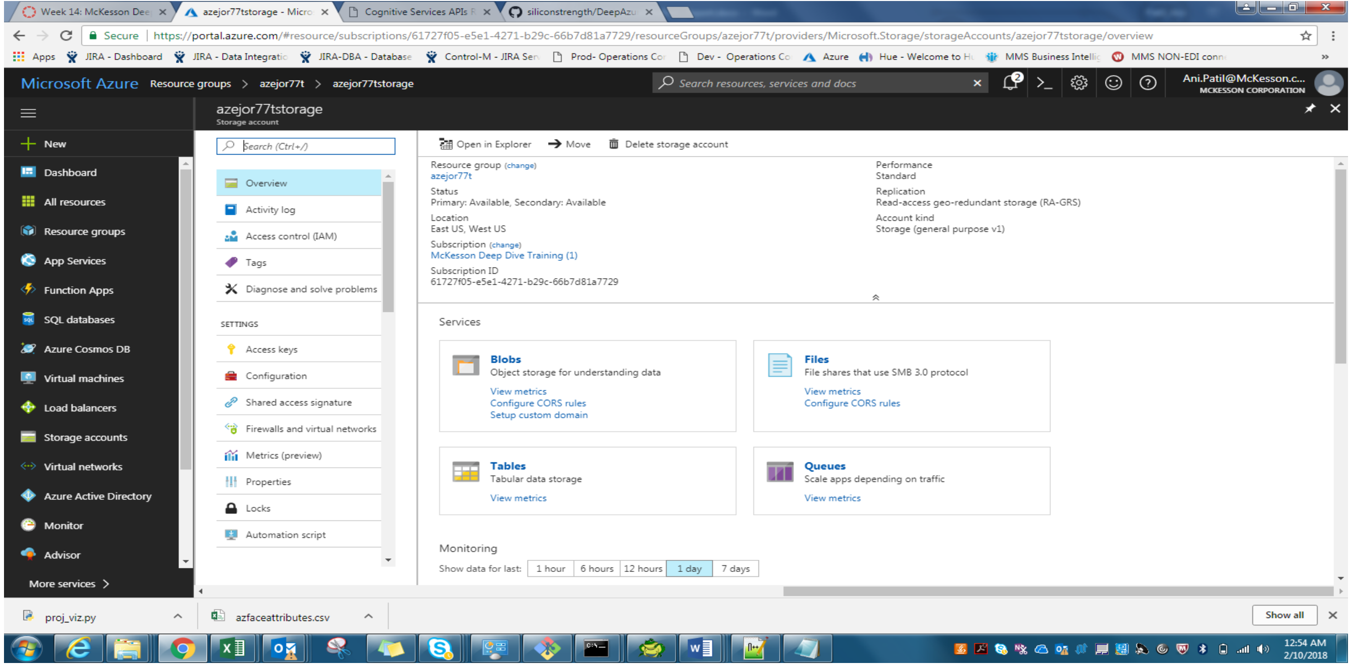
* Secure transfer required: Disabled | Enabled

* Subscription: McKesson Deep Dive Training (1)

* Resource group: ● Create new ○ Use existing

Create Azure Storage Table

- Get in Storage Account, click on Tables and add a new table



Resource group (change)
azejor77t

Status
Primary: Available, Secondary: Available

Location
East US, West US

Subscription (change)
McKesson Deep Dive Training (1)

Subscription ID
61727f05-e5e1-4271-b29c-66b7d81a7729

Performance
Standard

Replication
Read-access geo-redundant storage (RA-GRS)

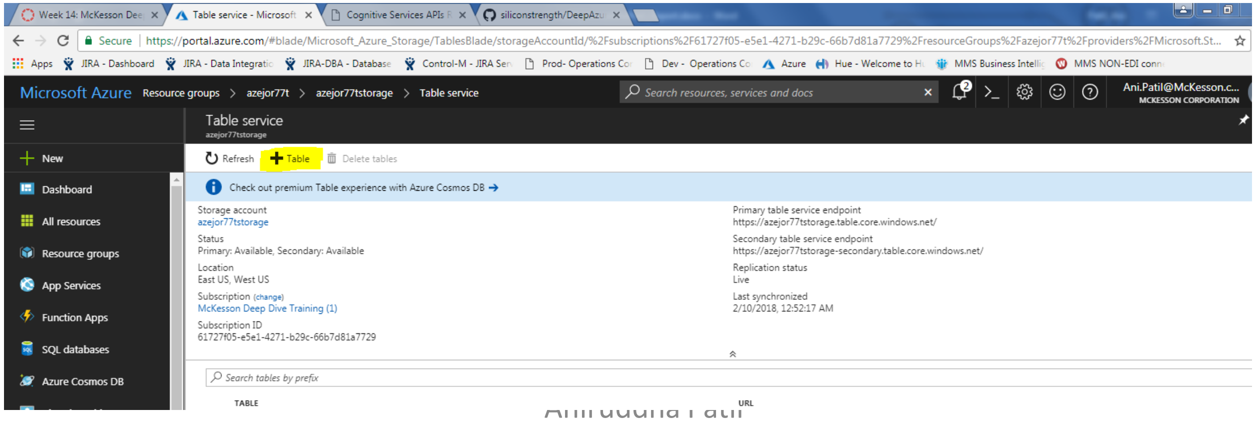
Account kind
Storage (general purpose v1)

Blobs
Object storage for understanding data
View metrics
Configure CORS rules
Setup custom domain

Files
File shares that use SMB 3.0 protocol
View metrics
Configure CORS rules

Tables
Tabular data storage
View metrics

Queues
Scale apps depending on traffic
View metrics



Storage account
azejor77tstorage

Status
Primary: Available, Secondary: Available

Location
East US, West US

Subscription (change)
McKesson Deep Dive Training (1)

Subscription ID
61727f05-e5e1-4271-b29c-66b7d81a7729

Primary table service endpoint
https://azejor77tstorage.table.core.windows.net/

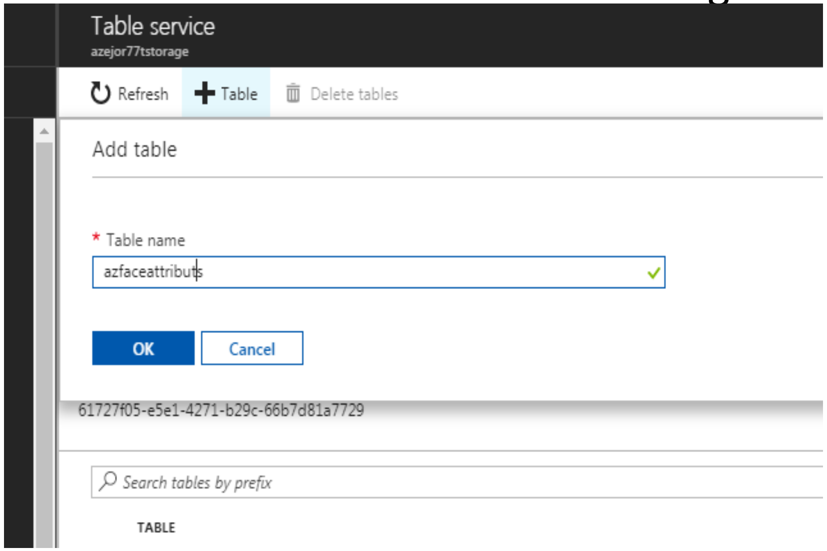
Secondary table service endpoint
https://azejor77tstorage-secondary.table.core.windows.net/

Replication status
Live

Last synchronized
2/10/2018, 12:52:17 AM

Create Azure Storage Table

- Give a name to table to see it created as below images

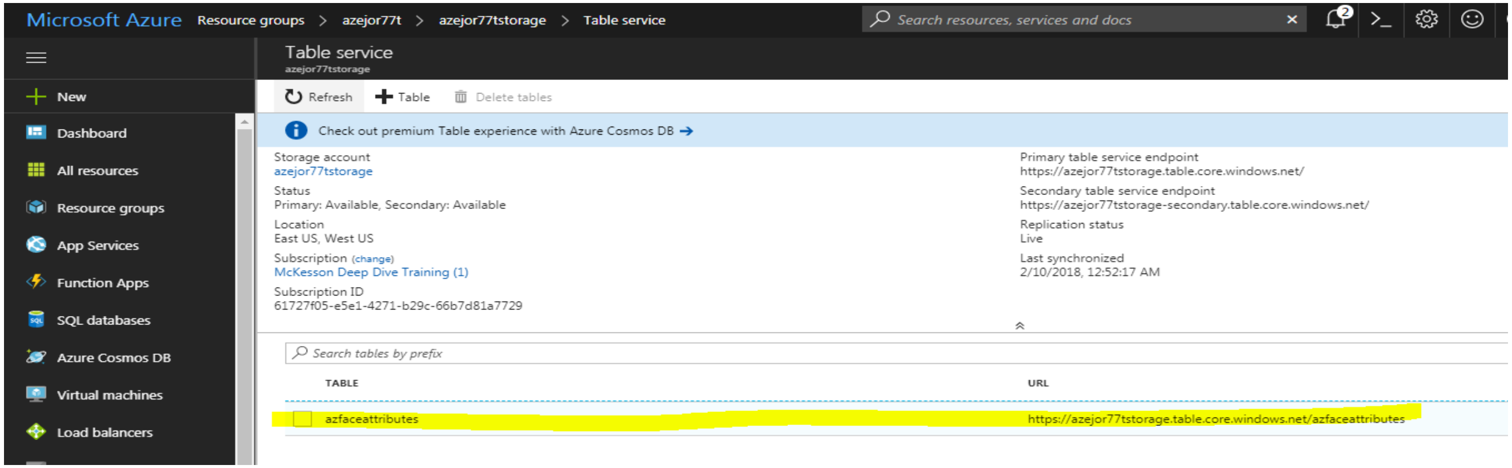


Add table

* Table name

azfaceattributs

61727f05-e5e1-4271-b29c-66b7d81a7729



Check out premium Table experience with Azure Cosmos DB →

Storage account
azejor77tstorage

Status
Primary: Available, Secondary: Available

Location
East US, West US

Subscription (change)
McKesson Deep Dive Training (1)

Subscription ID
61727f05-e5e1-4271-b29c-66b7d81a7729

Primary table service endpoint
https://azejor77tstorage.table.core.windows.net/

Secondary table service endpoint
https://azejor77tstorage-secondary.table.core.windows.net/

Replication status
Live

Last synchronized
2/10/2018, 12:52:17 AM

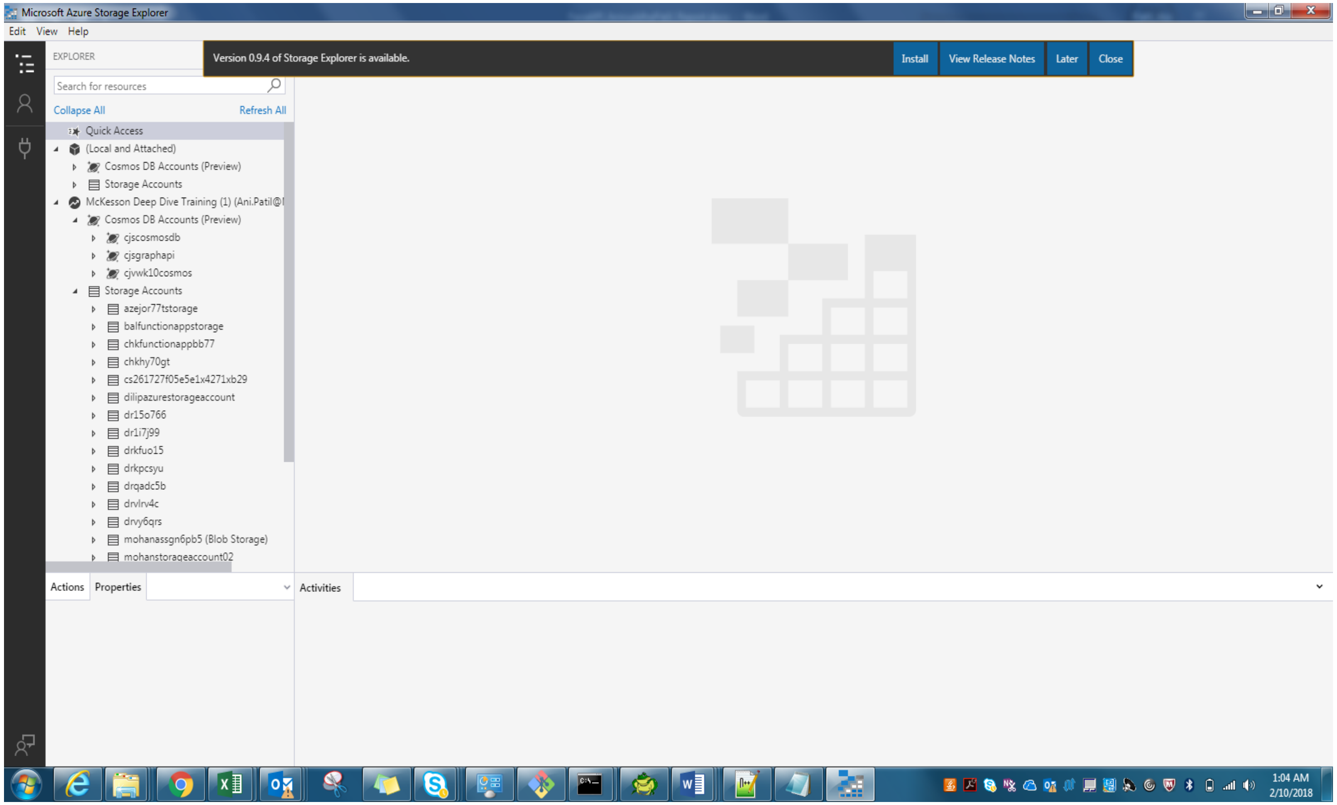
	TABLE	URL
□	azfaceattributes	https://azejor77tstorage.table.core.windows.net/azfaceattributes

Microsoft Azure Storage Explorer

- To access and manage this table, we will need to install – Microsoft Azure Storage Explorer. From link below-

https://azure.microsoft.com/en-us/features/storage-explorer/

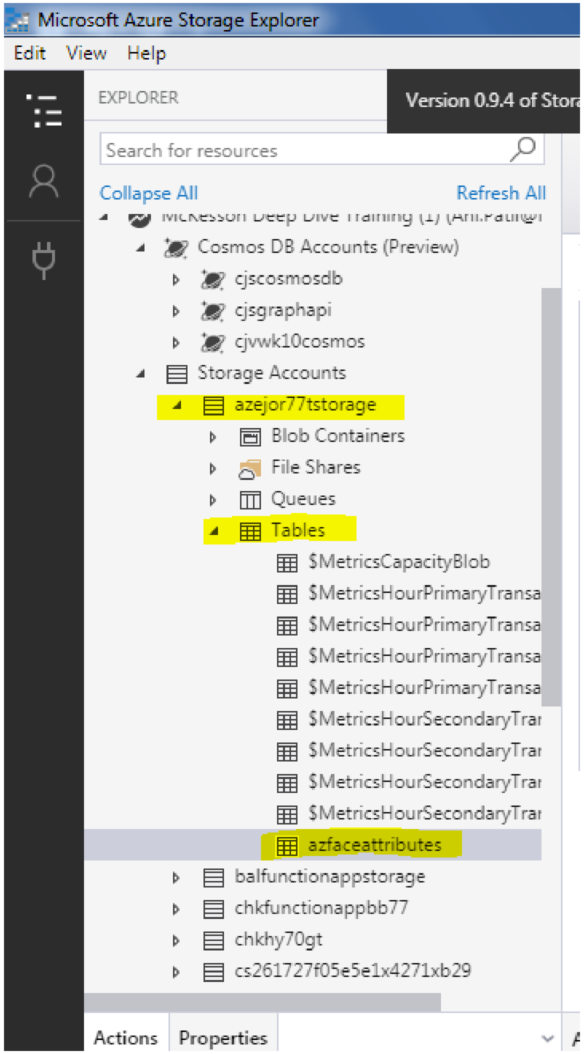
- Microsoft Azure Storage Explorer, when first opened, will ask for your Azure login credentials to connect to Azure storage



Version 0.9.4 of Storage Explorer is available.

Microsoft Azure Storage Explorer

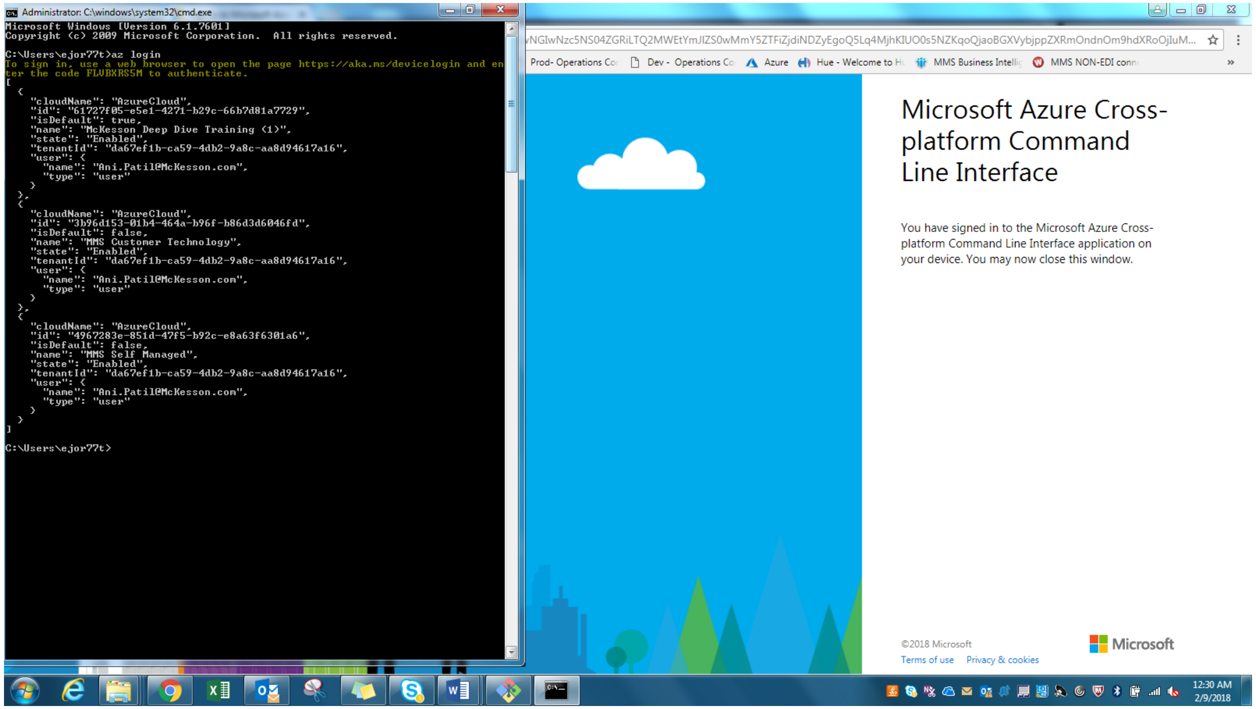
- Once you click your storage Account, it will display respective tables inside it



McKesson Deep Dive Training (1) (Ani.Patil@
Cosmos DB Accounts (Preview)
cjscosmosdb
cjsgraphapi
cjvwk10cosmos
Storage Accounts
azejor77tstorage
Blob Containers
File Shares
Queues
Tables
$MetricsCapacityBlob
$MetricsHourPrimaryTransa
$MetricsHourPrimaryTransa
$MetricsHourPrimaryTransa
$MetricsHourPrimaryTransa
$MetricsHourSecondaryTrar
$MetricsHourSecondaryTrar
$MetricsHourSecondaryTrar
$MetricsHourSecondaryTrar
azfaceattributes
balfunctionappstorage
chkfunctionappbb77
chkhy70gt
cs261727f05e5e1x4271xb29

Python & Azure CLI Login

- See that you have Python 3.6 installed.
- If not, download Anaconda Python 3.6 distribution.
- Login to Azure with Azure CLI



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\ejor77t>az login
To sign in, use a web browser to open the page https://aka.ms/devicelogin and en
ter the code FLVBXRS5M to authenticate.
[
  {
    "cloudName": "AzureCloud",
    "id": "61727f05-e5e1-4271-b29c-66b7d81a7729",
    "isDefault": true,
    "name": "McKesson Deep Dive Training (1)",
    "state": "Enabled",
    "tenantId": "da67ef1b-ca59-4db2-9a8c-aa8d94617a16",
    "user": {
      "name": "Ani.Patil@McKesson.com",
      "type": "user"
    }
  },
  {
    "cloudName": "AzureCloud",
    "id": "3b96d153-01b4-464a-b96f-b86d3d6046fd",
    "isDefault": false,
    "name": "MMS Customer Technology",
    "state": "Enabled",
    "tenantId": "da67ef1b-ca59-4db2-9a8c-aa8d94617a16",
    "user": {
      "name": "Ani.Patil@McKesson.com",
      "type": "user"
    }
  },
  {
    "cloudName": "AzureCloud",
    "id": "4967283e-851d-47f5-b92c-e8a63f6301a6",
    "isDefault": false,
    "name": "MMS Self Managed",
    "state": "Enabled",
    "tenantId": "da67ef1b-ca59-4db2-9a8c-aa8d94617a16",
    "user": {
      "name": "Ani.Patil@McKesson.com",
      "type": "user"
    }
  }
]

C:\Users\ejor77t>
```

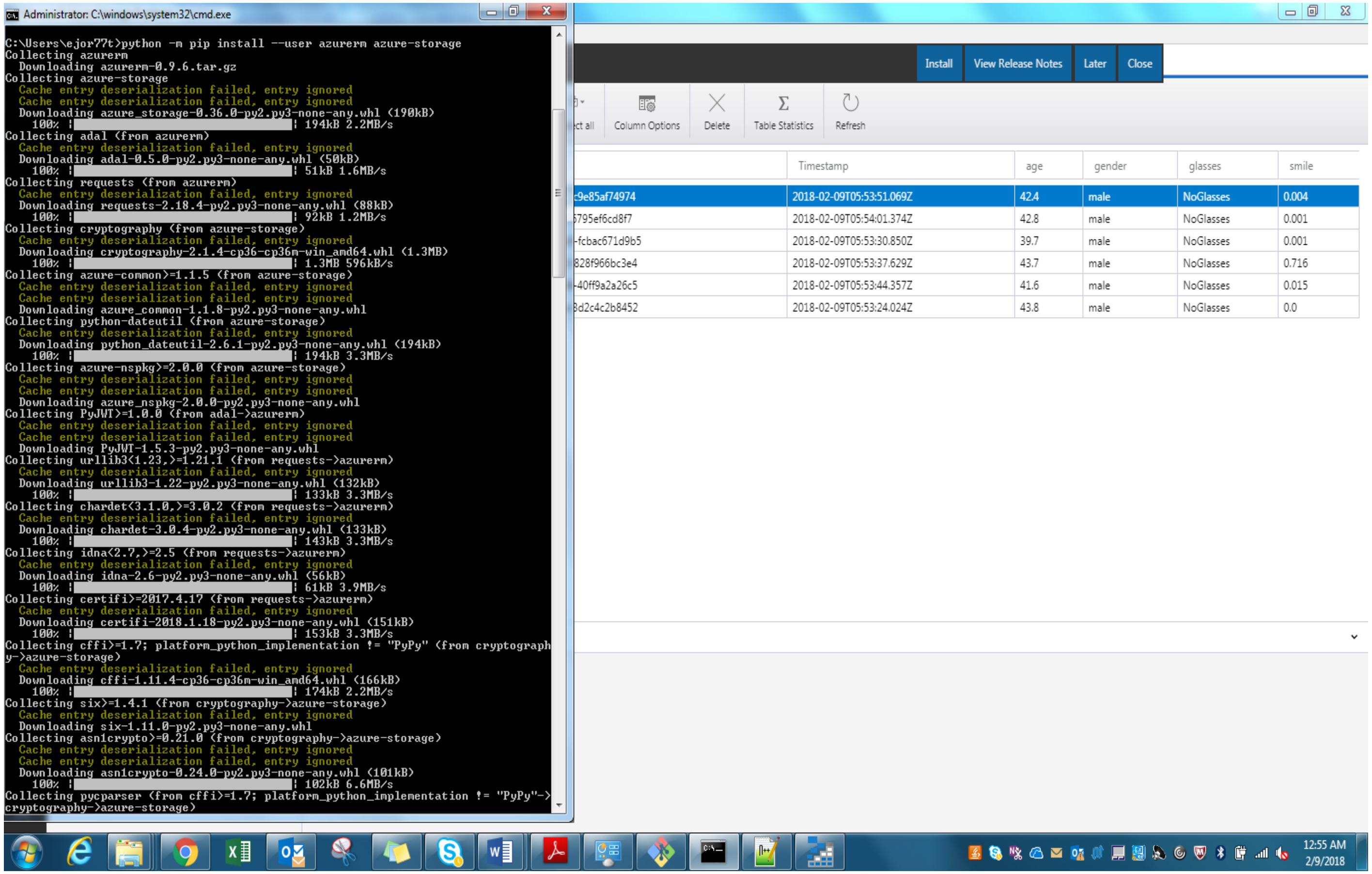
Microsoft Azure Cross-platform Command Line Interface

You have signed in to the Microsoft Azure Cross-platform Command Line Interface application on your device. You may now close this window.

©2018 Microsoft
Terms of use Privacy & cookies
Microsoft

Install Azure Storage Package

- Install storage package needed for working with Azure storage table



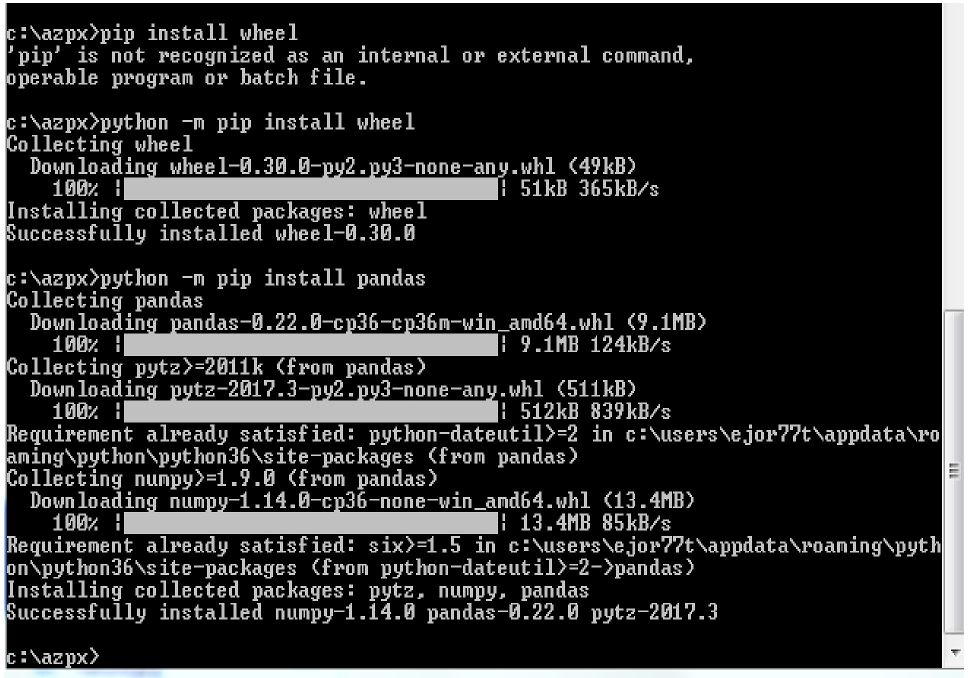
Administrator: C:\windows\system32\cmd.exe

```
C:\Users\ejor77t>python -m pip install --user azurerm azure-storage
Collecting azurerm
  Downloading azurerm-0.9.6.tar.gz
Collecting azure-storage
  Cache entry deserialization failed, entry ignored
  Cache entry deserialization failed, entry ignored
  Downloading azure_storage-0.36.0-py2.py3-none-any.whl (190kB)
    100% ¦████████████████████████████████¦ 194kB 2.2MB/s
Collecting adal (from azurerm)
  Cache entry deserialization failed, entry ignored
  Downloading adal-0.5.0-py2.py3-none-any.whl (50kB)
    100% ¦████████████████████████████████¦ 51kB 1.6MB/s
Collecting requests (from azurerm)
  Cache entry deserialization failed, entry ignored
  Downloading requests-2.18.4-py2.py3-none-any.whl (88kB)
    100% ¦████████████████████████████████¦ 92kB 1.2MB/s
Collecting cryptography (from azure-storage)
  Cache entry deserialization failed, entry ignored
  Downloading cryptography-2.1.4-cp36-cp36m-win_amd64.whl (1.3MB)
    100% ¦████████████████████████████████¦ 1.3MB 596kB/s
Collecting azure-common>=1.1.5 (from azure-storage)
  Cache entry deserialization failed, entry ignored
  Cache entry deserialization failed, entry ignored
  Downloading azure_common-1.1.8-py2.py3-none-any.whl
Collecting python-dateutil (from azure-storage)
  Cache entry deserialization failed, entry ignored
  Downloading python_dateutil-2.6.1-py2.py3-none-any.whl (194kB)
    100% ¦████████████████████████████████¦ 194kB 3.3MB/s
Collecting azure-nspkg>=2.0.0 (from azure-storage)
  Cache entry deserialization failed, entry ignored
  Cache entry deserialization failed, entry ignored
  Downloading azure_nspkg-2.0.0-py2.py3-none-any.whl
Collecting PyJWT>=1.0.0 (from adal->azurerm)
  Cache entry deserialization failed, entry ignored
  Cache entry deserialization failed, entry ignored
  Downloading PyJWT-1.5.3-py2.py3-none-any.whl
Collecting urllib3<1.23,>=1.21.1 (from requests->azurerm)
  Cache entry deserialization failed, entry ignored
  Downloading urllib3-1.22-py2.py3-none-any.whl (132kB)
    100% ¦████████████████████████████████¦ 133kB 3.3MB/s
Collecting chardet<3.1.0,>=3.0.2 (from requests->azurerm)
  Cache entry deserialization failed, entry ignored
  Downloading chardet-3.0.4-py2.py3-none-any.whl (133kB)
    100% ¦████████████████████████████████¦ 143kB 3.3MB/s
Collecting idna<2.7,>=2.5 (from requests->azurerm)
  Cache entry deserialization failed, entry ignored
  Downloading idna-2.6-py2.py3-none-any.whl (56kB)
    100% ¦████████████████████████████████¦ 61kB 3.9MB/s
Collecting certifi>=2017.4.17 (from requests->azurerm)
  Cache entry deserialization failed, entry ignored
  Downloading certifi-2018.1.18-py2.py3-none-any.whl (151kB)
    100% ¦████████████████████████████████¦ 153kB 3.3MB/s
Collecting cffi>=1.7; platform_python_implementation != "PyPy" (from cryptograph
y->azure-storage)
  Cache entry deserialization failed, entry ignored
  Downloading cffi-1.11.4-cp36-cp36m-win_amd64.whl (166kB)
    100% ¦████████████████████████████████¦ 174kB 2.2MB/s
Collecting six>=1.4.1 (from cryptography->azure-storage)
  Cache entry deserialization failed, entry ignored
  Downloading six-1.11.0-py2.py3-none-any.whl
Collecting asn1crypto>=0.21.0 (from cryptography->azure-storage)
  Cache entry deserialization failed, entry ignored
  Cache entry deserialization failed, entry ignored
  Downloading asn1crypto-0.24.0-py2.py3-none-any.whl (101kB)
    100% ¦████████████████████████████████¦ 102kB 6.6MB/s
Collecting pycparser (from cffi>=1.7; platform_python_implementation != "PyPy"->
cryptography->azure-storage)
```

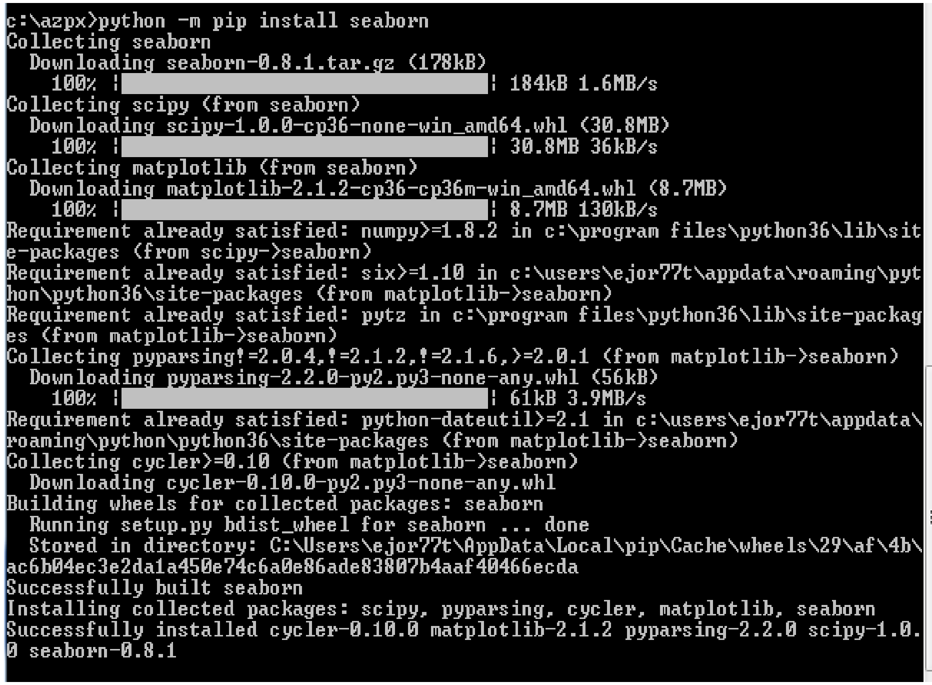
	Timestamp	age	gender	glasses	smile
c9e85af74974	2018-02-09T05:53:51.069Z	42.4	male	NoGlasses	0.004
5795ef6cd8f7	2018-02-09T05:54:01.374Z	42.8	male	NoGlasses	0.001
-fcbac671d9b5	2018-02-09T05:53:30.850Z	39.7	male	NoGlasses	0.001
828f966bc3e4	2018-02-09T05:53:37.629Z	43.7	male	NoGlasses	0.716
-40ff9a2a26c5	2018-02-09T05:53:44.357Z	41.6	male	NoGlasses	0.015
3d2c4c2b8452	2018-02-09T05:53:24.024Z	43.8	male	NoGlasses	0.0

Install Pandas

- Install wheel, pandas, seaborn – This we will need later for visualization



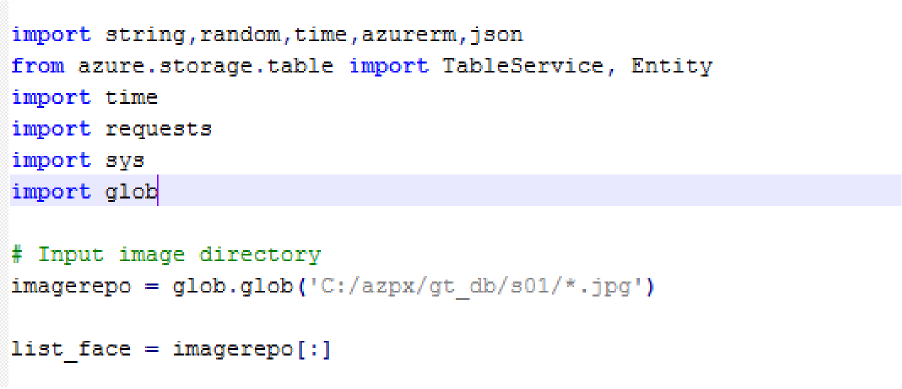
```
c:\azpx>pip install wheel
'pip' is not recognized as an internal or external command,
operable program or batch file.

c:\azpx>python -m pip install wheel
Collecting wheel
  Downloading wheel-0.30.0-py2.py3-none-any.whl (49kB)
    100% |████████████████████████████████| 51kB 365kB/s
Installing collected packages: wheel
Successfully installed wheel-0.30.0

c:\azpx>python -m pip install pandas
Collecting pandas
  Downloading pandas-0.22.0-cp36-cp36m-win_amd64.whl (9.1MB)
    100% |████████████████████████████████| 9.1MB 124kB/s
Collecting pytz>=2011k (from pandas)
  Downloading pytz-2017.3-py2.py3-none-any.whl (511kB)
    100% |████████████████████████████████| 512kB 839kB/s
Requirement already satisfied: python-dateutil>=2 in c:\users\ejor77t\appdata\ro
aming\python\python36\site-packages (from pandas)
Collecting numpy>=1.9.0 (from pandas)
  Downloading numpy-1.14.0-cp36-none-win_amd64.whl (13.4MB)
    100% |████████████████████████████████| 13.4MB 85kB/s
Requirement already satisfied: six>=1.5 in c:\users\ejor77t\appdata\roaming\pyth
on\python36\site-packages (from python-dateutil>=2->pandas)
Installing collected packages: pytz, numpy, pandas
Successfully installed numpy-1.14.0 pandas-0.22.0 pytz-2017.3

c:\azpx>
```



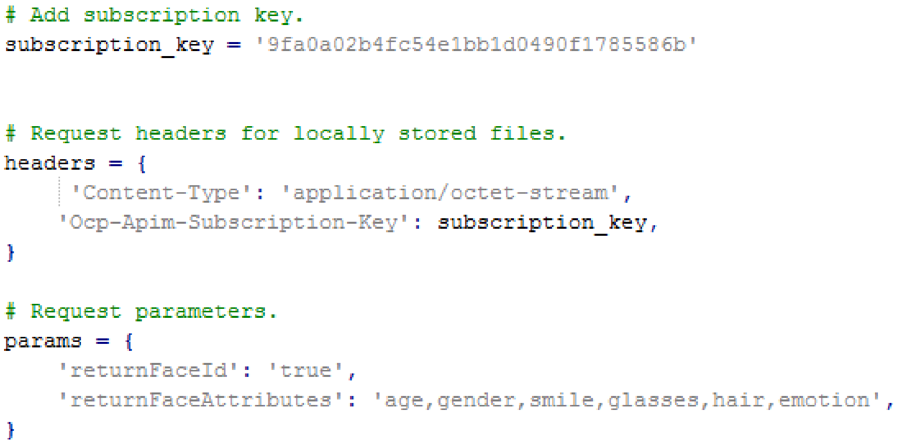
```
c:\azpx>python -m pip install seaborn
Collecting seaborn
  Downloading seaborn-0.8.1.tar.gz (178kB)
    100% |████████████████████████████████| 184kB 1.6MB/s
Collecting scipy (from seaborn)
  Downloading scipy-1.0.0-cp36-none-win_amd64.whl (30.8MB)
    100% |████████████████████████████████| 30.8MB 36kB/s
Collecting matplotlib (from seaborn)
  Downloading matplotlib-2.1.2-cp36-cp36m-win_amd64.whl (8.7MB)
    100% |████████████████████████████████| 8.7MB 130kB/s
Requirement already satisfied: numpy>=1.8.2 in c:\program files\python36\lib\sit
e-packages (from scipy->seaborn)
Requirement already satisfied: six>=1.10 in c:\users\ejor77t\appdata\roaming\pyt
hon\python36\site-packages (from matplotlib->seaborn)
Requirement already satisfied: pytz in c:\program files\python36\lib\site-packag
es (from matplotlib->seaborn)
Collecting pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 (from matplotlib->seaborn)
  Downloading pyparsing-2.2.0-py2.py3-none-any.whl (56kB)
    100% |████████████████████████████████| 61kB 3.9MB/s
Requirement already satisfied: python-dateutil>=2.1 in c:\users\ejor77t\appdata\
roaming\python\python36\site-packages (from matplotlib->seaborn)
Collecting cycler>=0.10 (from matplotlib->seaborn)
  Downloading cycler-0.10.0-py2.py3-none-any.whl
Building wheels for collected packages: seaborn
  Running setup.py bdist_wheel for seaborn ... done
  Stored in directory: C:\Users\ejor77t\AppData\Local\pip\Cache\wheels\29\af\4b\
ac6b04ec3e2da1a450e74c6a0e86ade83807b4aaf40466ecda
Successfully built seaborn
Installing collected packages: scipy, pyparsing, cycler, matplotlib, seaborn
Successfully installed cycler-0.10.0 matplotlib-2.1.2 pyparsing-2.2.0 scipy-1.0.
0 seaborn-0.8.1
```

Face API Demonstration

- Face API call and do some face detection and get face attributes
- Below is code snippet of my Face API program- azfaceapi.py
- Setting up subset of input images pulled from above Face Database to be processed by Face API-



```
import string,random,time,azurerm,json
from azure.storage.table import TableService, Entity
import time
import requests
import sys
import glob

# Input image directory
imagerepo = glob.glob('C:/azpx/gt_db/s01/*.jpg')

list_face = imagerepo[:]
```

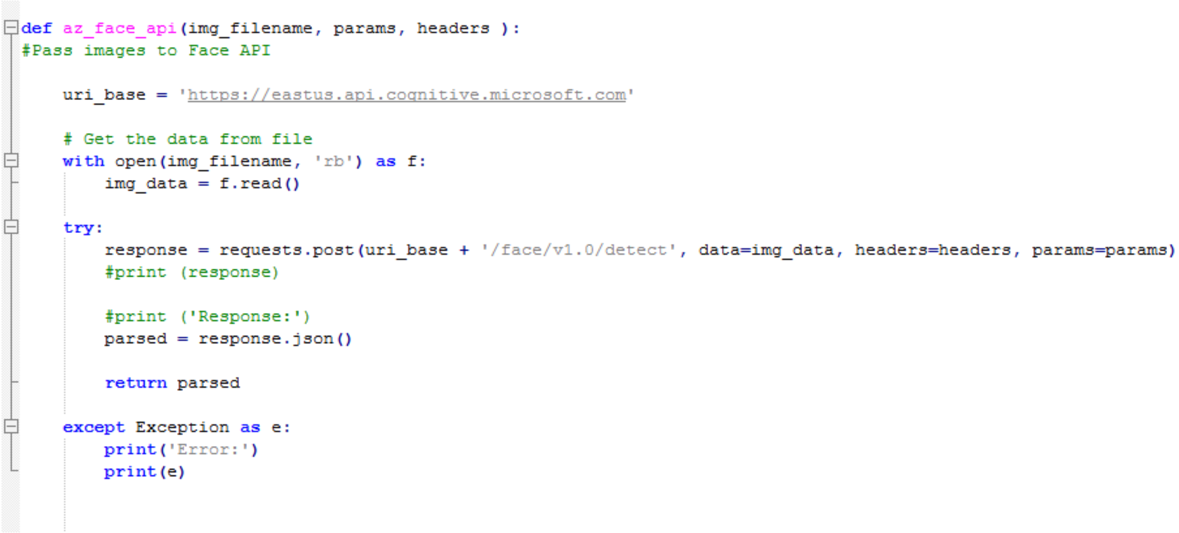


```
# Add subscription key.
subscription_key = '9fa0a02b4fc54e1bb1d0490f1785586b'


# Request headers for locally stored files.
headers = {
    'Content-Type': 'application/octet-stream',
    'Ocp-Apim-Subscription-Key': subscription_key,
}

# Request parameters.
params = {
    'returnFaceId': 'true',
    'returnFaceAttributes': 'age,gender,smile,glasses,hair,emotion',
}
```

Set subscription key for face API and header and params for request

Face API Demonstration

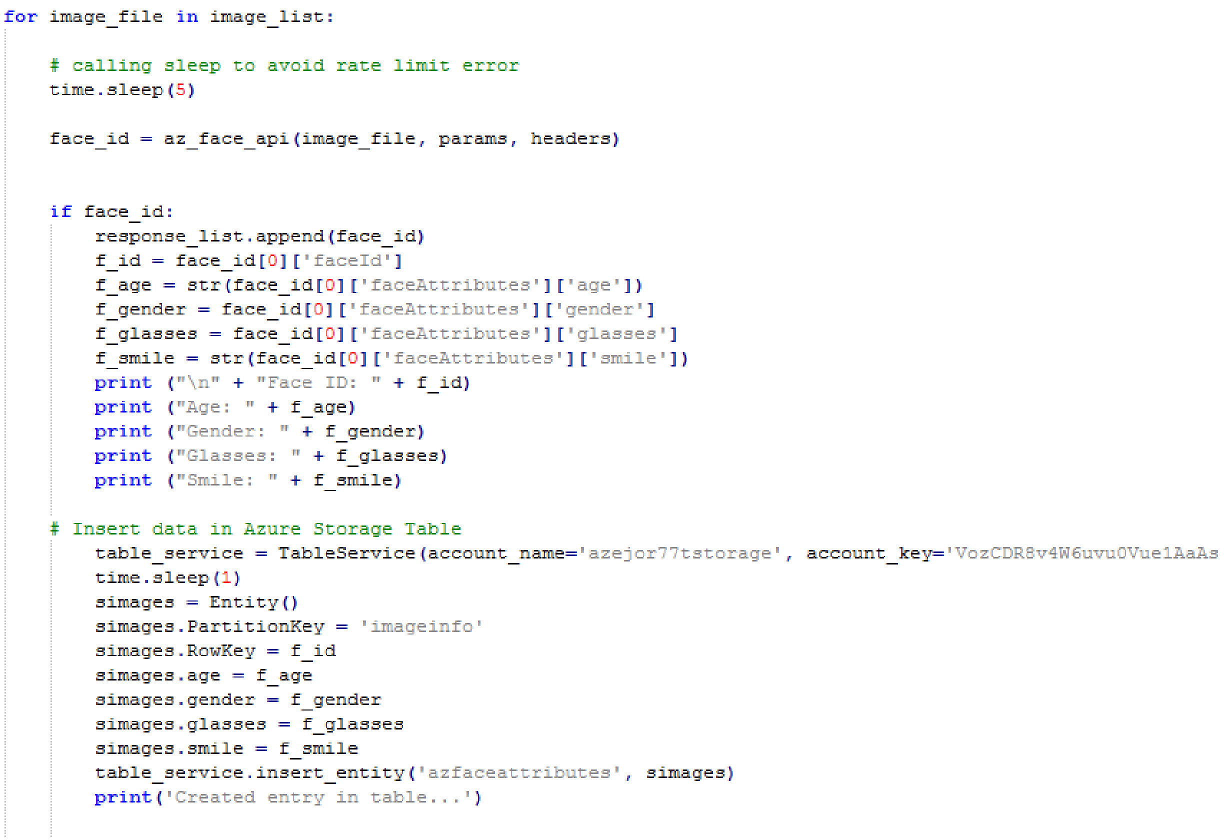
- Call Face API for each of input image to analyze and collect face attributes



```
def az_face_api(img_filename, params, headers ):
#Pass images to Face API

    uri_base = 'https://eastus.api.cognitive.microsoft.com'

    # Get the data from file
    with open(img_filename, 'rb') as f:
        img_data = f.read()

    try:
        response = requests.post(uri_base + '/face/v1.0/detect', data=img_data, headers=headers, params=params)
        #print (response)

        #print ('Response:')
        parsed = response.json()

        return parsed

    except Exception as e:
        print('Error:')
        print(e)
```

Store Face Attributes to Azure Storage

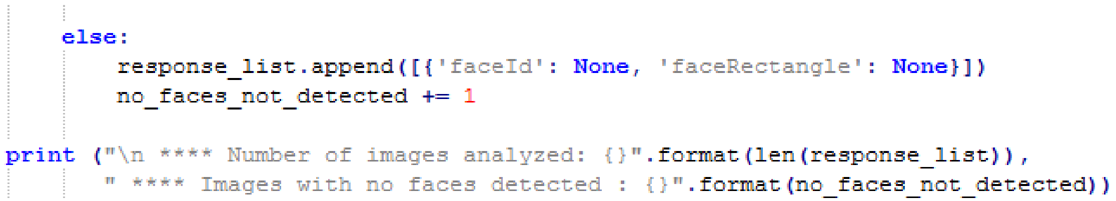
- Once we get requested face attributes, result will be in JSON format. We will collect respective values and store them in Azure Storage table as below



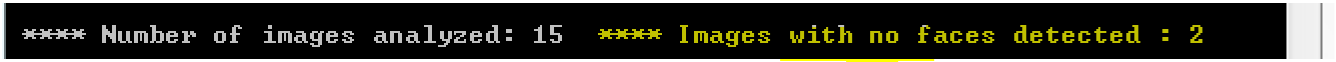
```
for image_file in image_list:

    # calling sleep to avoid rate limit error
    time.sleep(5)

    face_id = az_face_api(image_file, params, headers)


    if face_id:
        response_list.append(face_id)
        f_id = face_id[0]['faceId']
        f_age = str(face_id[0]['faceAttributes']['age'])
        f_gender = face_id[0]['faceAttributes']['gender']
        f_glasses = face_id[0]['faceAttributes']['glasses']
        f_smile = str(face_id[0]['faceAttributes']['smile'])
        print ("\n" + "Face ID: " + f_id)
        print ("Age: " + f_age)
        print ("Gender: " + f_gender)
        print ("Glasses: " + f_glasses)
        print ("Smile: " + f_smile)

    # Insert data in Azure Storage Table
        table_service = TableService(account_name='azejor77tstorage', account_key='VozCDR8v4W6uvu0Vue1AaAs
        time.sleep(1)
        simages = Entity()
        simages.PartitionKey = 'imageinfo'
        simages.RowKey = f_id
        simages.age = f_age
        simages.gender = f_gender
        simages.glasses = f_glasses
        simages.smile = f_smile
        table_service.insert_entity('azfaceattributes', simages)
        print('Created entry in table...')
```

Validate results for images with No Face

- We will also be checking for images with no faces in it and will count such images as below.
- Snapshot of program output below shows 2 images with no face detected.



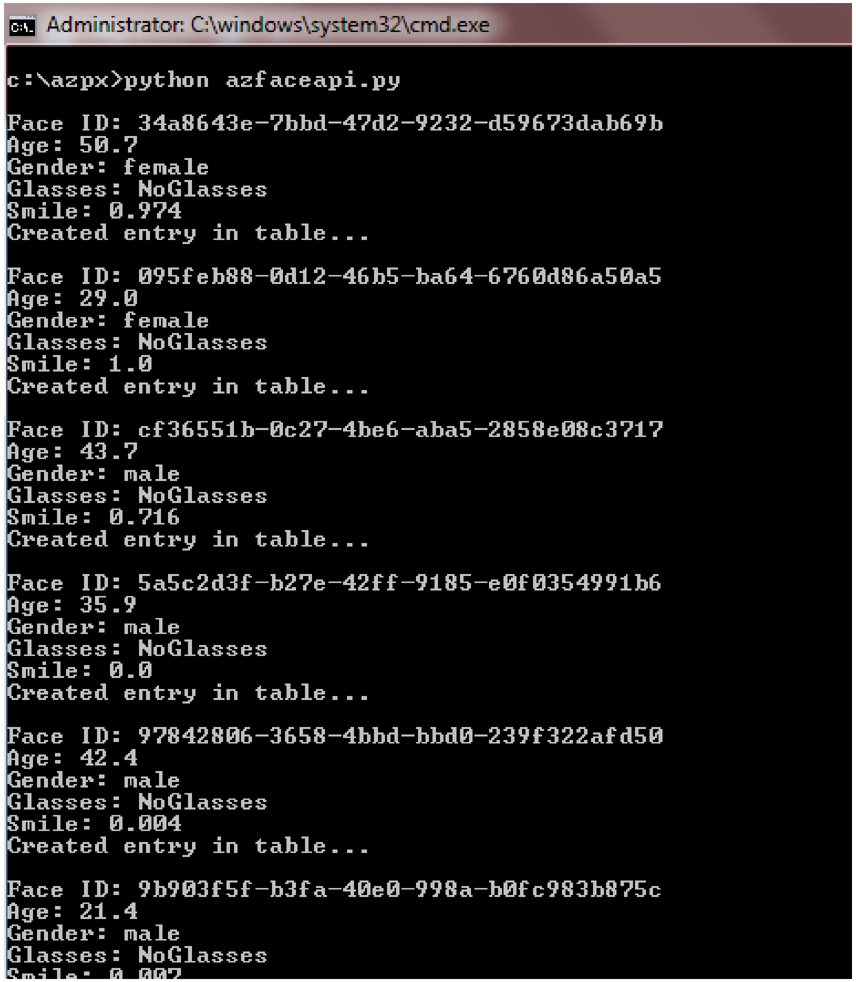
```
    else:
        response_list.append([{'faceId': None, 'faceRectangle': None}])
        no_faces_not_detected += 1

print ("\n **** Number of images analyzed: {}".format(len(response_list)),
       " **** Images with no faces detected : {}".format(no_faces_not_detected))
```



```
***** Number of images analyzed: 15  ***** Images with no faces detected : 2
```

Run Face API Program

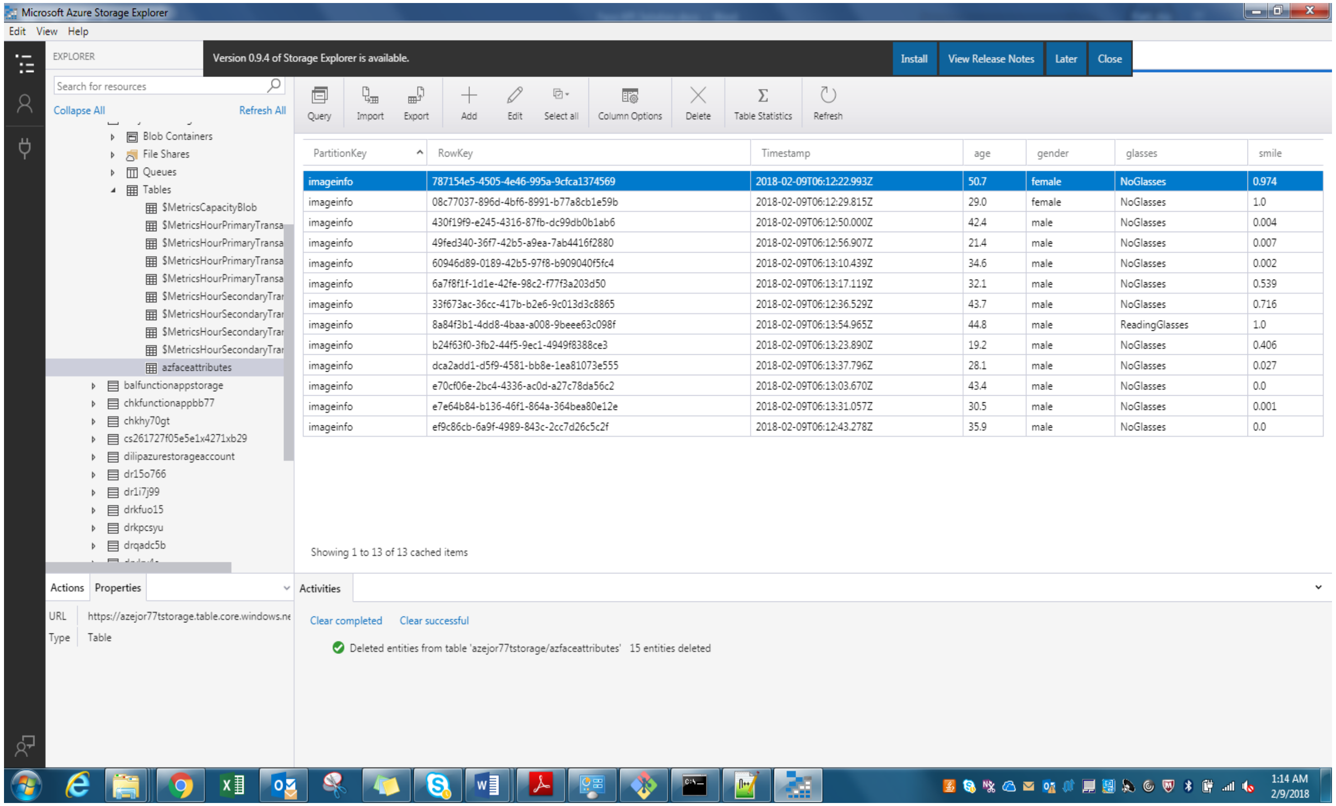
- We will also be checking for images with no faces in it and will count such images as below



```
c:\azpx>python azfaceapi.py

Face ID: 34a8643e-7bbd-47d2-9232-d59673dab69b
Age: 50.7
Gender: female
Glasses: NoGlasses
Smile: 0.974
Created entry in table...

Face ID: 095feb88-0d12-46b5-ba64-6760d86a50a5
Age: 29.0
Gender: female
Glasses: NoGlasses
Smile: 1.0
Created entry in table...

Face ID: cf36551b-0c27-4be6-aba5-2858e08c3717
Age: 43.7
Gender: male
Glasses: NoGlasses
Smile: 0.716
Created entry in table...

Face ID: 5a5c2d3f-b27e-42ff-9185-e0f0354991b6
Age: 35.9
Gender: male
Glasses: NoGlasses
Smile: 0.0
Created entry in table...

Face ID: 97842806-3658-4bbd-bbd0-239f322afd50
Age: 42.4
Gender: male
Glasses: NoGlasses
Smile: 0.004
Created entry in table...

Face ID: 9b903f5f-b3fa-40e0-998a-b0fc983b875c
Age: 21.4
Gender: male
Glasses: NoGlasses
Smile: 0.002
```

Face API Results stored in Azure Storage Table

- Output of Face Detection Analysis is stored in Azure Table Storage and when looked through Microsoft Azure Storage Explorer will look as below-



Version 0.9.4 of Storage Explorer is available.

PartitionKey	RowKey	Timestamp	age	gender	glasses	smile
imageinfo	787154e5-4505-4e46-995a-9cfca1374569	2018-02-09T06:12:22.993Z	50.7	female	NoGlasses	0.974
imageinfo	08c77037-896d-4bf6-8991-b77a8cb1e59b	2018-02-09T06:12:29.815Z	29.0	female	NoGlasses	1.0
imageinfo	430f19f9-e245-4316-87fb-dc99db0b1ab6	2018-02-09T06:12:50.000Z	42.4	male	NoGlasses	0.004
imageinfo	49fed340-36f7-42b5-a9ea-7ab4416f2880	2018-02-09T06:12:56.907Z	21.4	male	NoGlasses	0.007
imageinfo	60946d89-0189-42b5-97f8-b909040f5fc4	2018-02-09T06:13:10.439Z	34.6	male	NoGlasses	0.002
imageinfo	6a7f8f1f-1d1e-42fe-98c2-f77f3a203d50	2018-02-09T06:13:17.119Z	32.1	male	NoGlasses	0.539
imageinfo	33f673ac-36cc-417b-b2e6-9c013d3c8865	2018-02-09T06:12:36.529Z	43.7	male	NoGlasses	0.716
imageinfo	8a84f3b1-4dd8-4baa-a008-9beee63c098f	2018-02-09T06:13:54.965Z	44.8	male	ReadingGlasses	1.0
imageinfo	b24f63f0-3fb2-44f5-9ec1-4949f8388ce3	2018-02-09T06:13:23.890Z	19.2	male	NoGlasses	0.406
imageinfo	dca2add1-d5f9-4581-bb8e-1ea81073e555	2018-02-09T06:13:37.796Z	28.1	male	NoGlasses	0.027
imageinfo	e70cf06e-2bc4-4336-ac0d-a27c78da56c2	2018-02-09T06:13:03.670Z	43.4	male	NoGlasses	0.0
imageinfo	e7e64b84-b136-46f1-864a-364bea80e12e	2018-02-09T06:13:31.057Z	30.5	male	NoGlasses	0.001
imageinfo	ef9c86cb-6a9f-4989-843c-2cc7d26c5c2f	2018-02-09T06:12:43.278Z	35.9	male	NoGlasses	0.0

Showing 1 to 13 of 13 cached items

URL	https://azejor77tstorage.table.core.windows.ne
Type	Table

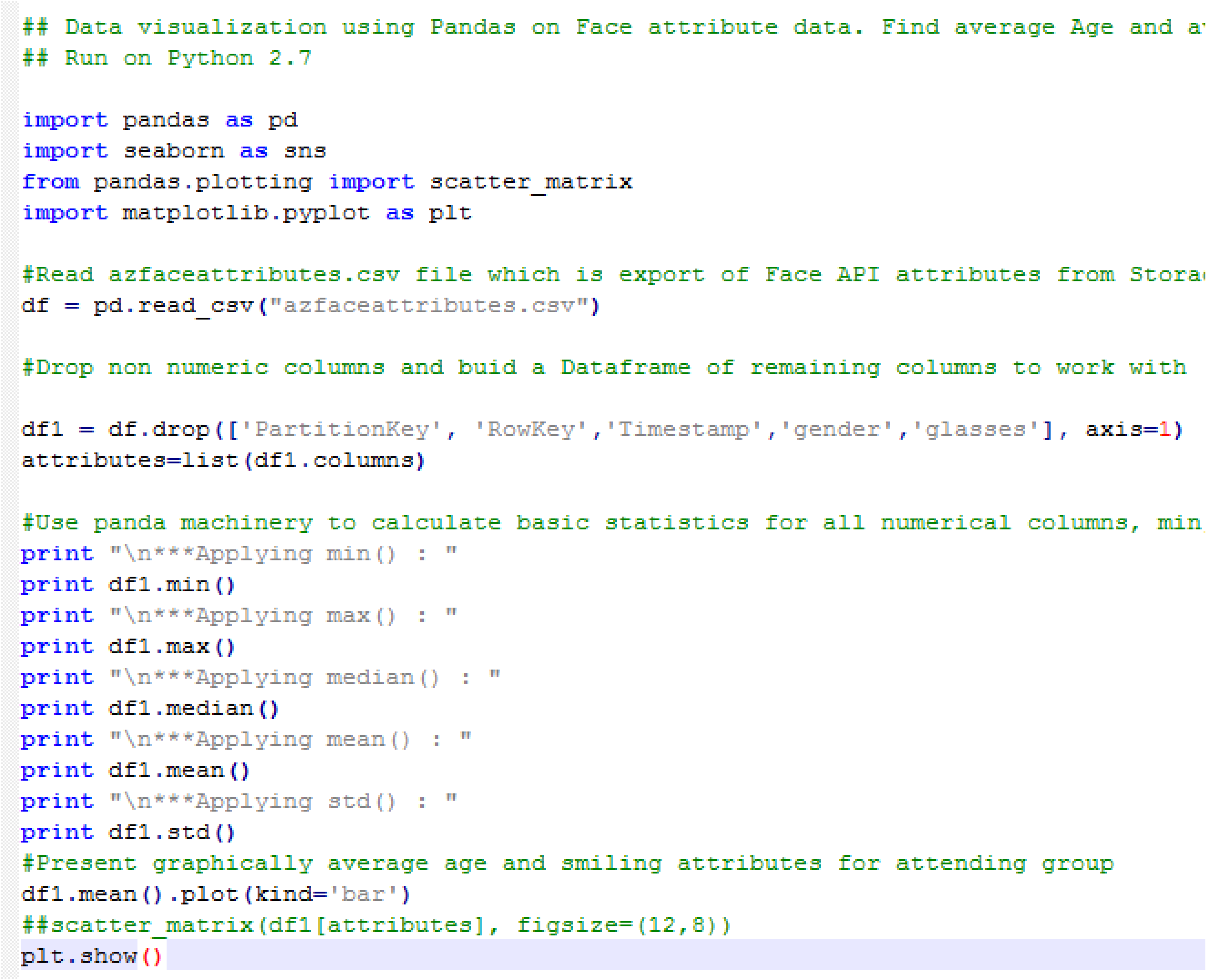
Deleted entities from table 'azejor77tstorage/azfaceattributes' 15 entities deleted

Data Visualization

- Run some data visualization on output data in above storage table. Please extract this table data as csv file – azfaceattributes.csv
- Now we will use Pandas to run visualization. I have created python program proj_viz.py for same.
- Python 3.6 on windows 7 machine was giving error on some of python pandas package installation so I ran visualization using Python 2.7 on Windows 10 machine.
- In this python code, I have calculated some statistical figures covering group of faces analyzed. It pulls minimum, maximum, average, mean, median age and smile factor of group.
- Visualization shows average age and average smiling face attribute for group of faces analyzed. I created Bar chart showing these averages.

Data Visualization

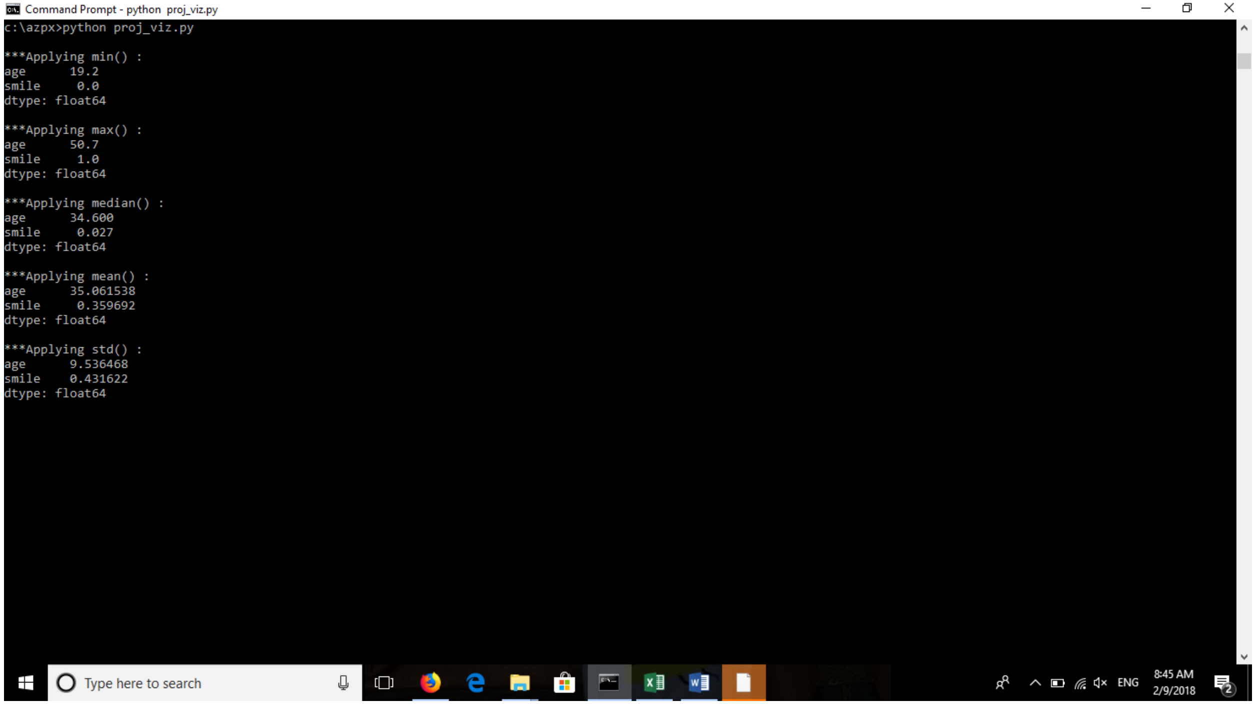
- Code snippet



```
## Data visualization using Pandas on Face attribute data. Find average Age and a
## Run on Python 2.7

import pandas as pd
import seaborn as sns
from pandas.plotting import scatter_matrix
import matplotlib.pyplot as plt

#Read azfaceattributes.csv file which is export of Face API attributes from Stora
df = pd.read_csv("azfaceattributes.csv")

#Drop non numeric columns and buid a Dataframe of remaining columns to work with

df1 = df.drop(['PartitionKey', 'RowKey','Timestamp','gender','glasses'], axis=1)
attributes=list(df1.columns)

#Use panda machinery to calculate basic statistics for all numerical columns, min
print "\n***Applying min() : "
print df1.min()
print "\n***Applying max() : "
print df1.max()
print "\n***Applying median() : "
print df1.median()
print "\n***Applying mean() : "
print df1.mean()
print "\n***Applying std() : "
print df1.std()
#Present graphically average age and smiling attributes for attending group
df1.mean().plot(kind='bar')
##scatter_matrix(df1[attributes], figsize=(12,8))
plt.show()
```

Data Visualization

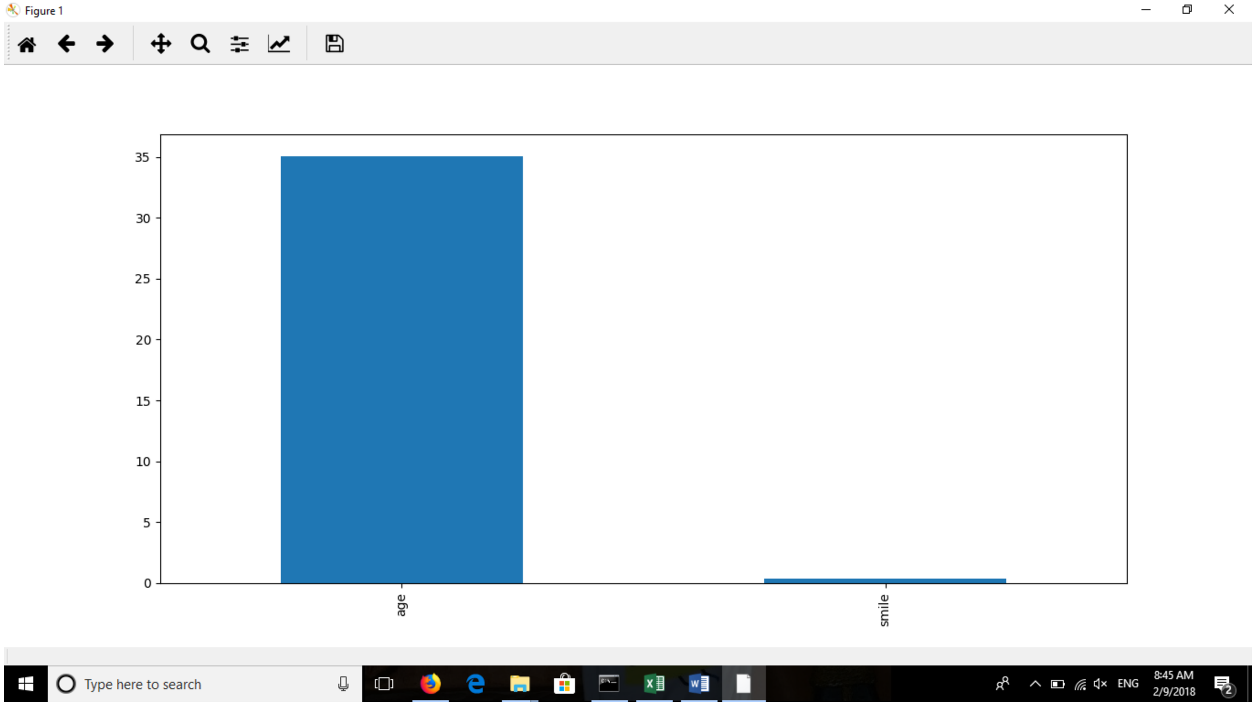
- Program will show data visualization as bar chart for average age and smile face attribute for all image faces for input data. Such data can be used for deriving social intelligence from public events.



```
c:\azpx>python proj_viz.py

***Applying min() :
age      19.2
smile     0.0
dtype: float64

***Applying max() :
age      50.7
smile     1.0
dtype: float64

***Applying median() :
age      34.600
smile     0.027
dtype: float64

***Applying mean() :
age      35.061538
smile     0.359692
dtype: float64

***Applying std() :
age      9.536468
smile    0.431622
dtype: float64
```

Data Visualization

- You can see that average age of group is 34.6 and Smile factor of 0.35.



YouTube URLs, GitHub URL, Last Page

- Two minute (short): https://www.youtube.com/watch?v=l_3xt2chxns
- 15 minutes (long): https://www.youtube.com/watch?v=7dBOHjJYPj4
- GitHub Repository with all artifacts: https://github.com/siliconstrength/DeepAzureFinalProject