1/6/2019 Intro to GCP!

Intro to Google Cloud Platform. This notebook illustrates using built-in machine learning models from GCP in 5 steps



Pre-requisite: Install GCP for python using the following pip command

- · pip install google-cloud
- · pip install google-cloud-vision
- · pip install google-cloud-language

Example 1: Image Content Analysis

• By using Cloud Vision API from GCP the content of the image is analysed

```
In [ ]: # Step 0: Import statements
    from google.cloud import vision
    from google.cloud.vision import types
    import io
    import os
    "Packages Imported successfully"
To [ ]: # Step 1: Service-hev Authentication
```

```
In [ ]: # Step 1: Service-key Authentication
    os.environ["GOOGLE_APPLICATION_CREDENTIALS"] = r"C:/Users/kmy07/Desktop/ML/servi
    ce-key.json"
```

```
In [ ]: # Step 2: Intialize Client Object to request service
    client_object = vision.ImageAnnotatorClient()
```

1/6/2019 Intro to GCP!

```
In [ ]: # Step 3: Read the Image
        imagePath = r"C:\Users\kmy07\Desktop\ML\InputImages\leaves.jpg"
        with io.open(imagePath, 'rb') as inputImage:
            requestContent = inputImage.read()
        print("Input Image\n")
        from IPython.display import Image
        Image(imagePath,width = 300,height = 300)
In [ ]: #Step 4: Request-Reply Preparation
        #Prepare request
        request = types.Image(content = requestContent)
        #Perform service request and get Response
        response = client object.label detection(image=request)
In [ ]: #Step 5: Handle the response
        labels = response.label_annotations
        print('Labels:')
        for label in labels:
            print(label.description)
```

Example 2: Sentiment Analysis

· By using Cloud Natural Language API from GCP the sentiment of the statement is predicted

```
In [ ]: # Step 0: Import statements
    from google.cloud import language
    from google.cloud.language import enums
    from google.cloud.language import types

"Packages Imported successfully"

In [ ]: # Step 1: Service-key Authentication
    os.environ["GOOGLE_APPLICATION_CREDENTIALS"] = r"C:/Users/kmy07/Desktop/ML/service-key.json"

In [ ]: # Step 2: Intialize Client Object to request service
    client_object = language.LanguageServiceClient()
```

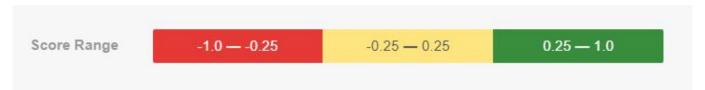
1/6/2019 Intro to GCP!

```
In [ ]: # Step 3: Read the input text
        inputText = u"The Great Living Chola Temples were built by kings of the Chola Em
        pire, which stretched over all of south India and the neighbouring islands. The
         site includes three great 11th- and 12th-century Temples: the Brihadisvara Temp
        le at Thanjavur, the Brihadisvara Temple at Gangaikondacholisvaram and the Airav
        atesvara Temple at Darasuram. The Temple of Gangaikondacholisvaram, built by Raj
        endra I, was completed in 1035. Its 53-m vimana (sanctum tower) has recessed cor
        ners and a graceful upward curving movement, contrasting with the straight and s
        evere tower at Thanjavur. The Airavatesvara temple complex, built by Rajaraja I
        I, at Darasuram features a 24-m vimana and a stone image of Shiva. The temples t
        estify to the brilliant achievements of the Chola in architecture, sculpture, pa
        inting and bronze casting."
        document = types.Document(
            content=inputText,
            type=enums.Document.Type.PLAIN_TEXT)
In [ ]: # Step 4.1:
        # Detects the sentiment of the text
        sentiment = client_object.analyze_sentiment(document=document).document_sentimen
        print('Text: {}\n'.format(inputText))
        print('Sentiment: {}'.format(sentiment.score))
In [ ]: # Step 4.2:
        # Analyse each entity
        entities = client_object.analyze_entities(document).entities
        for entity in entities:
            entity_type = enums.Entity.Type(entity.type)
```

Reference: Sentiment Index as given by google

print('=' * 20)

rl', '-')))



print(u'{:<16}: {}'.format('wikipedia_url', entity.metadata.get('wikipedia_u</pre>

print(u'{:<16}: {}'.format('mid', entity.metadata.get('mid', '-')))</pre>

print(u'{:<16}: {}'.format('name', entity.name))
print(u'{:<16}: {}'.format('type', entity_type.name))
print(u'{:<16}: {}'.format('salience', entity.salience))</pre>