Handwritten Scripts Classifier

USING GOOGLE CLOUD AUTOML AND APIS

Names	SRN
Shubham S	PES1UG20CS420
Shashank S	PES1UG20CS394

Objective

The goal of our application is to take in handwritten text images written by students and to give out the text and convert this into speech in any language the evaluator prefers. Natural Language processing is also done on this text to identify entities and other features.

A custom dataset which has sentences classified into 3 labels (Physics , Chemistry , Biology) is taken and trained using AutoML . The text identified from the image is given to this model which finally classifies it to the appropriate subject.

Pretrained APIs used

- Cloud Vision

  To identify the Text present in the Image.

- Text to speech

  To convert the text we obtained from the image to an audio file.

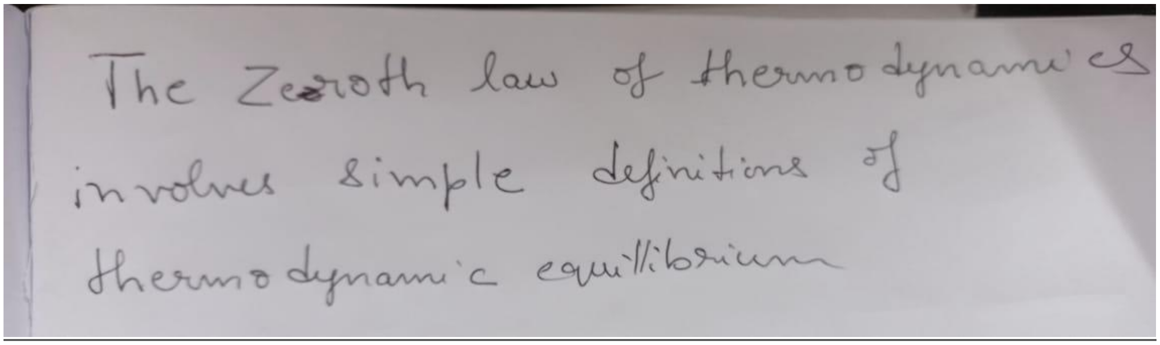
- Translate

  To translate the text we obtained to any language chosen.

- Natural language

  To analyze the text read.

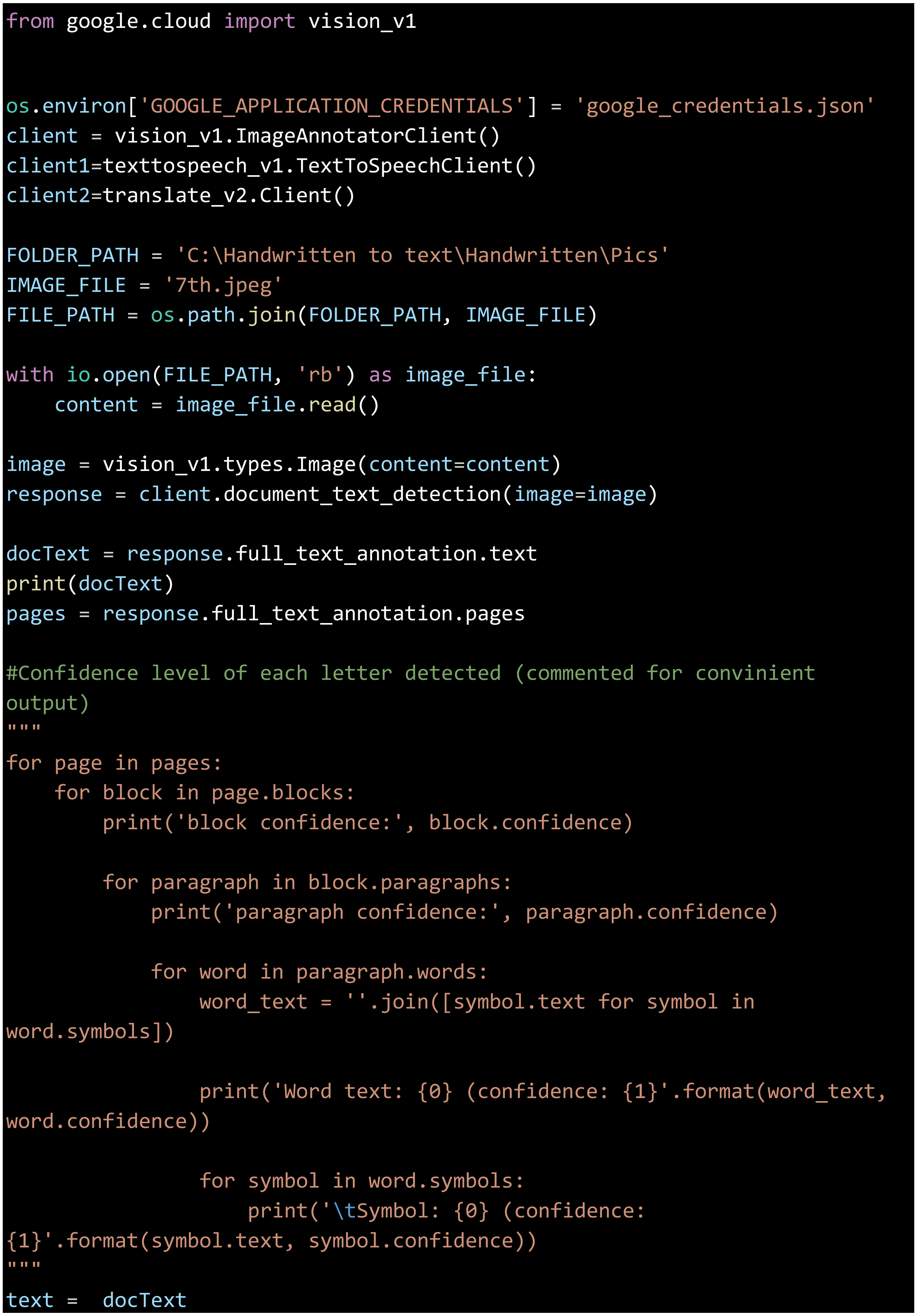
Sample Input



The Zeroth law of thermodynamics
involves simple definitions of
thermodynamic equillibrium

Code

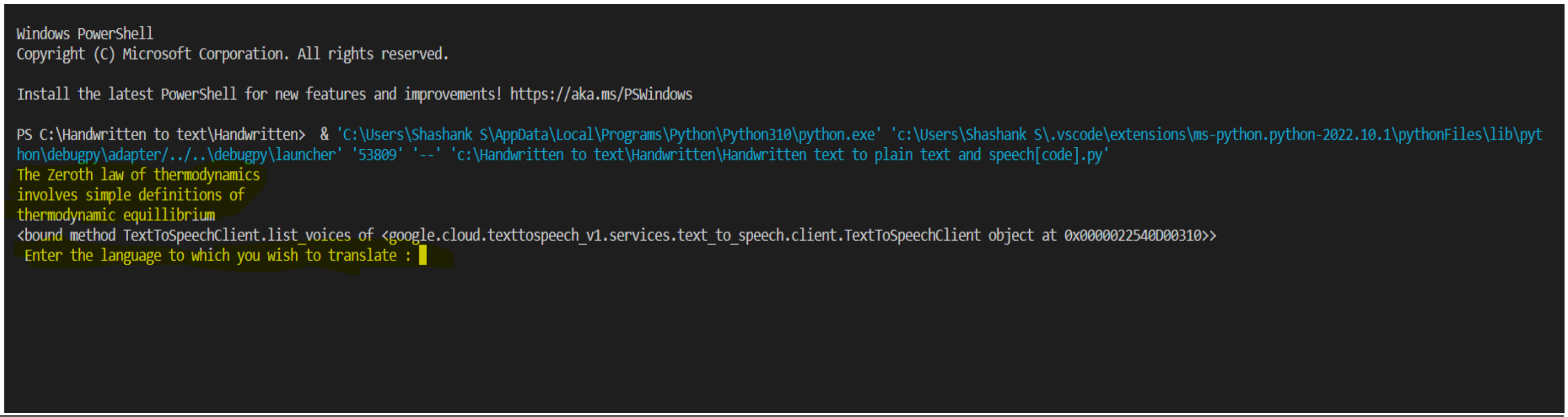
```
import os, io
import pandas as pd

from google.cloud import texttospeech_v1
from google.cloud import translate_v2
from google.cloud import language_v1
```



```
from google.cloud import vision_v1


os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = 'google_credentials.json'
client = vision_v1.ImageAnnotatorClient()
client1=texttospeech_v1.TextToSpeechClient()
client2=translate_v2.Client()

FOLDER_PATH = 'C:\Handwritten to text\Handwritten\Pics'
IMAGE_FILE = '7th.jpeg'
FILE_PATH = os.path.join(FOLDER_PATH, IMAGE_FILE)

with io.open(FILE_PATH, 'rb') as image_file:
    content = image_file.read()

image = vision_v1.types.Image(content=content)
response = client.document_text_detection(image=image)

docText = response.full_text_annotation.text
print(docText)
pages = response.full_text_annotation.pages

#Confidence level of each letter detected (commented for convinient
output)
"""
for page in pages:
    for block in page.blocks:
        print('block confidence:', block.confidence)

        for paragraph in block.paragraphs:
            print('paragraph confidence:', paragraph.confidence)

            for word in paragraph.words:
                word_text = ''.join([symbol.text for symbol in
word.symbols])

                print('Word text: {0} (confidence: {1}'.format(word_text,
word.confidence))

                for symbol in word.symbols:
                    print('\tSymbol: {0} (confidence:
{1}'.format(symbol.text, symbol.confidence))
"""
text =  docText
```

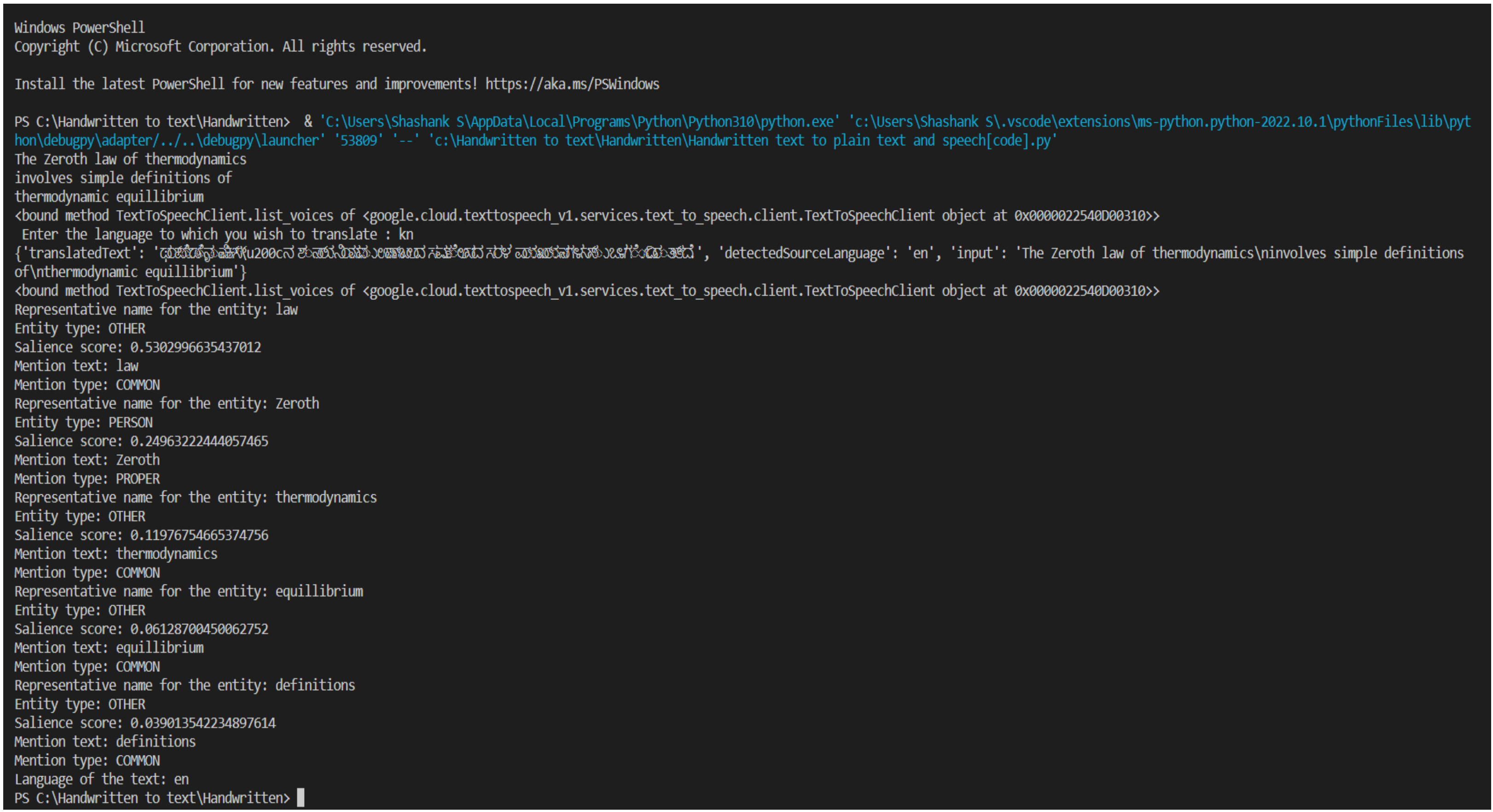
```
synthesis_input = texttospeech_v1.SynthesisInput(ssml=text)

voice1 = texttospeech_v1.VoiceSelectionParams(
    language_code = 'en-in',
    ssml_gender= texttospeech_v1.SsmlVoiceGender.FEMALE )


print(client1.list_voices)
audio_config= texttospeech_v1.AudioConfig(
    audio_encoding = texttospeech_v1.AudioEncoding.MP3
)

response1 = client1.synthesize_speech (
    input=synthesis_input,
    voice=voice1,
    audio_config=audio_config
)

with open('audio.mp3','wb',) as output:
    output.write(response1.audio_content)


target = input(" Enter the language to which you wish to translate : ")

output1 = client2.translate(text,target_language = target)

print(output1)


synthesis_input1 = 
texttospeech_v1.SynthesisInput(ssml=output1['translatedText'])

voice2 = texttospeech_v1.VoiceSelectionParams(
    language_code = target,
    ssml_gender= texttospeech_v1.SsmlVoiceGender.FEMALE )


print(client1.list_voices)
audio_config= texttospeech_v1.AudioConfig(
    audio_encoding = texttospeech_v1.AudioEncoding.MP3
)

response2 = client1.synthesize_speech (
    input=synthesis_input1,
    voice=voice2,
    audio_config=audio_config
)

with open('audio1.mp3','wb',) as output10:
    output10.write(response2.audio_content)



def sample_analyze_entities(docText):
    client3 = language_v1.LanguageServiceClient()
    type_ = language_v1.Document.Type.PLAIN_TEXT
    language = "en"
    document = {"content": docText, "type_": type_, "language": language}
    encoding_type = language_v1.EncodingType.UTF8
    response = client3.analyze_entities(request = {'document': document,
'encoding_type': encoding_type})
    for entity in response.entities:
        print(u"Representative name for the entity:
{}".format(entity.name))
        print(u"Entity type:
{}".format(language_v1.Entity.Type(entity.type_).name))
        print(u"Salience score: {}".format(entity.salience))
        for metadata_name, metadata_value in entity.metadata.items():
            print(u"{}: {}".format(metadata_name, metadata_value))
        for mention in entity.mentions:
            print(u"Mention text: {}".format(mention.text.content))
            print(u"Mention type:
{}".format(language_v1.EntityMention.Type(mention.type_).name)  )
    print(u"Language of the text: {}".format(response.language))

sample_analyze_entities(docText)
```

Identified Text



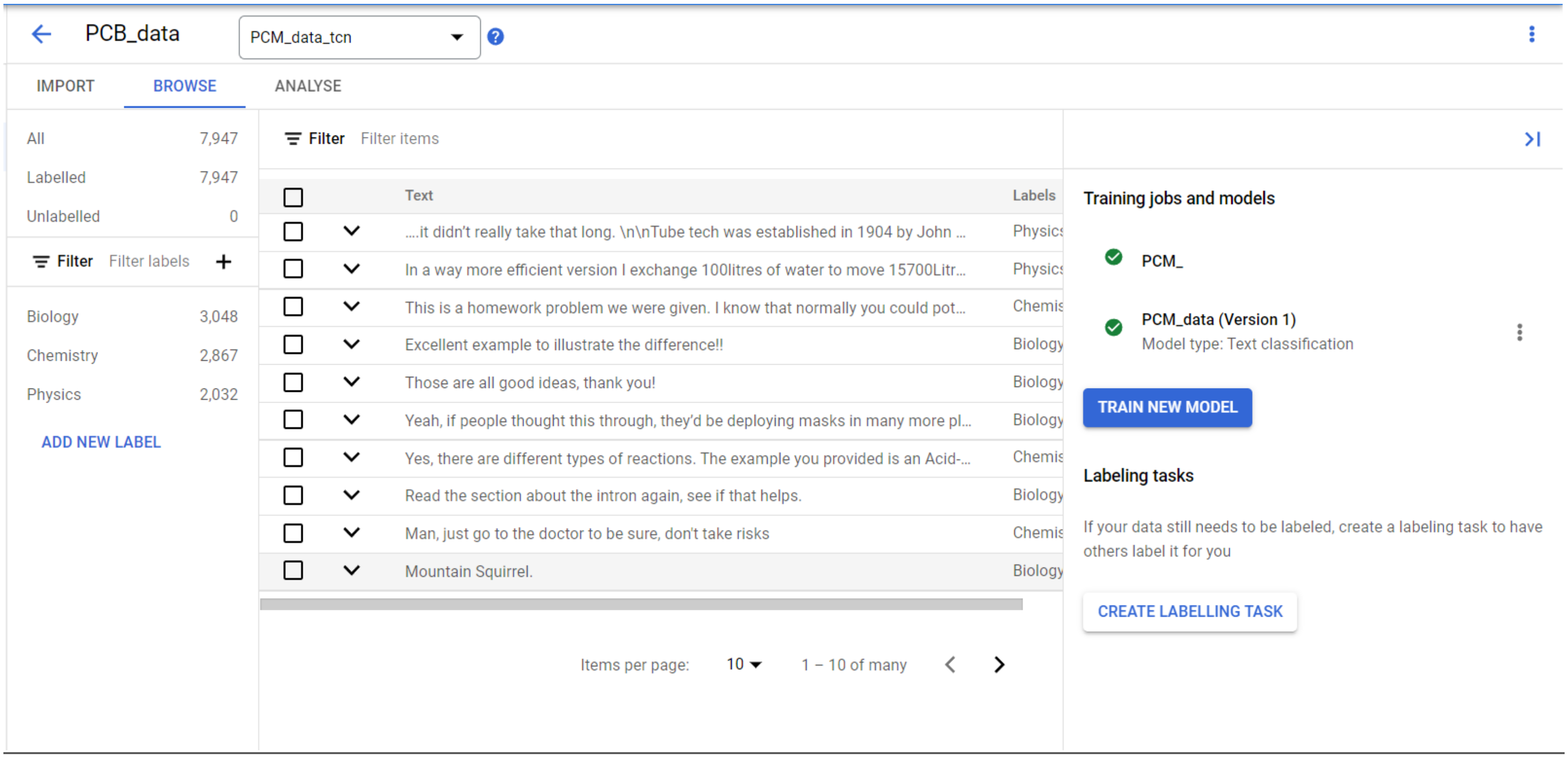
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Handwritten to text\Handwritten>  & 'C:\Users\Shashank S\AppData\Local\Programs\Python\Python310\python.exe' 'c:\Users\Shashank S\.vscode\extensions\ms-python.python-2022.10.1\pythonFiles\lib\pyt
hon\debugpy\adapter/../..\debugpy\launcher' '53809' '--' 'c:\Handwritten to text\Handwritten\Handwritten text to plain text and speech[code].py'
The Zeroth law of thermodynamics
involves simple definitions of
thermodynamic equillibrium
<bound method TextToSpeechClient.list_voices of <google.cloud.texttospeech_v1.services.text_to_speech.client.TextToSpeechClient object at 0x0000022540D00310>>
 Enter the language to which you wish to translate : 
```

Converting Text to different language (both text and to speech) with NLP analysis .



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Handwritten to text\Handwritten>  & 'C:\Users\Shashank S\AppData\Local\Programs\Python\Python310\python.exe' 'c:\Users\Shashank S\.vscode\extensions\ms-python.python-2022.10.1\pythonFiles\lib\pyt
hon\debugpy\adapter/../..\debugpy\launcher' '53809' '--' 'c:\Handwritten to text\Handwritten\Handwritten text to plain text and speech[code].py'
The Zeroth law of thermodynamics
involves simple definitions of
thermodynamic equillibrium
<bound method TextToSpeechClient.list_voices of <google.cloud.texttospeech_v1.services.text_to_speech.client.TextToSpeechClient object at 0x0000022540D00310>>
 Enter the language to which you wish to translate : kn
{'translatedText': 'ಥರ್ಮೋಡೈನಾಮಿಕ್ಸ್\u200cನ ಶೂನ್ಯ ನಿಯಮವು ಉಷ್ಣಬಲದ ಸಮತೋಲನದ ಸರಳ ವ್ಯಾಖ್ಯಾನಗಳನ್ನು ಒಳಗೊಂಡಿರುತ್ತದೆ', 'detectedSourceLanguage': 'en', 'input': 'The Zeroth law of thermodynamics\ninvolves simple definitions
of\nthermodynamic equillibrium'}
<bound method TextToSpeechClient.list_voices of <google.cloud.texttospeech_v1.services.text_to_speech.client.TextToSpeechClient object at 0x0000022540D00310>>
Representative name for the entity: law
Entity type: OTHER
Salience score: 0.5302996635437012
Mention text: law
Mention type: COMMON
Representative name for the entity: Zeroth
Entity type: PERSON
Salience score: 0.24963222444057465
Mention text: Zeroth
Mention type: PROPER
Representative name for the entity: thermodynamics
Entity type: OTHER
Salience score: 0.11976754665374756
Mention text: thermodynamics
Mention type: COMMON
Representative name for the entity: equillibrium
Entity type: OTHER
Salience score: 0.06128700450062752
Mention text: equillibrium
Mention type: COMMON
Representative name for the entity: definitions
Entity type: OTHER
Salience score: 0.039013542234897614
Mention text: definitions
Mention type: COMMON
Language of the text: en
PS C:\Handwritten to text\Handwritten> 
```

AutoML Dataset



All	7,947
Labelled	7,947
Unlabelled	0
Biology	3,048
Chemistry	2,867
Physics	2,032

	Text	Labels
□	....it didn't really take that long. \n\nTube tech was established in 1904 by John ...	Physic
□	In a way more efficient version I exchange 100litres of water to move 15700Litr...	Physic
□	This is a homework problem we were given. I know that normally you could pot...	Chemis
□	Excellent example to illustrate the difference!!	Biology
□	Those are all good ideas, thank you!	Biology
□	Yeah, if people thought this through, they'd be deploying masks in many more pl...	Biology
□	Yes, there are different types of reactions. The example you provided is an Acid-...	Chemis
□	Read the section about the intron again, see if that helps.	Biology
□	Man, just go to the doctor to be sure, don't take risks	Chemis
□	Mountain Squirrel.	Biology

Items per page: 10 1 – 10 of many

Training jobs and models

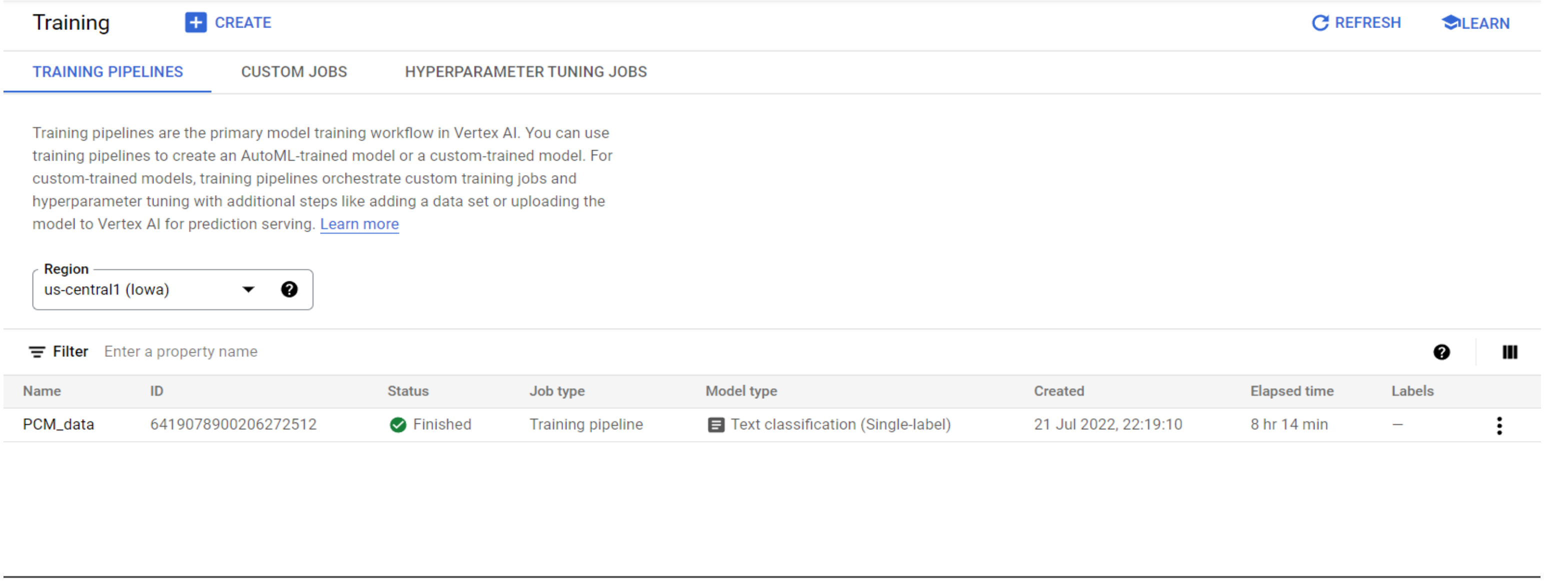
✓ PCM_

✓ PCM_data (Version 1)
Model type: Text classification

Labeling tasks

If your data still needs to be labeled, create a labeling task to have others label it for you

Training

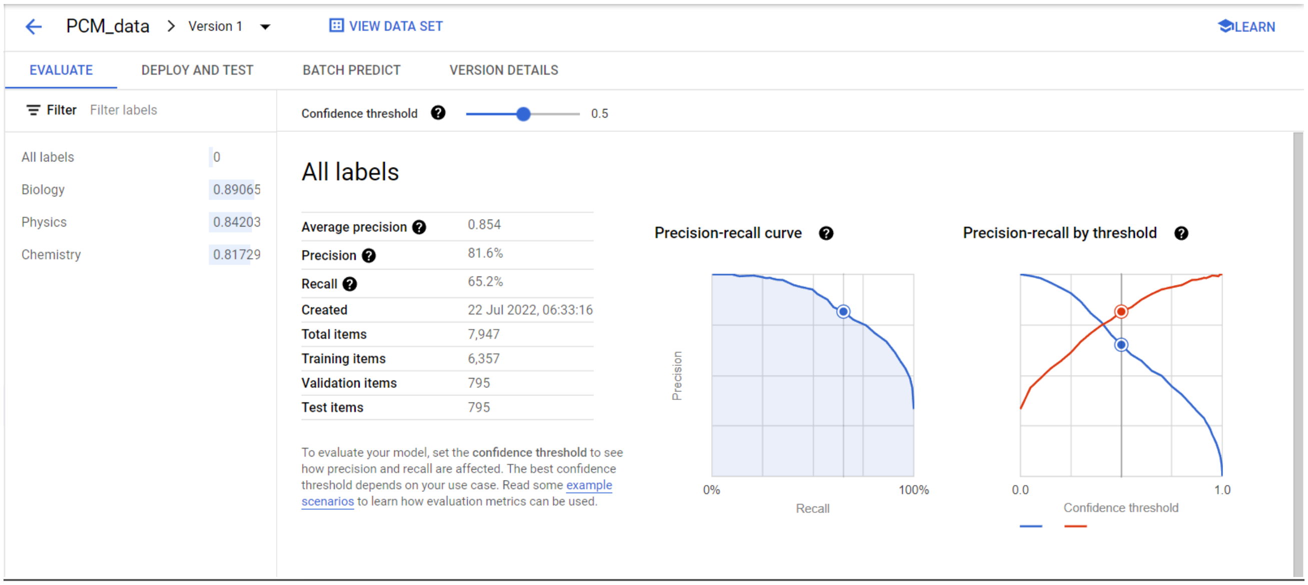


Training pipelines are the primary model training workflow in Vertex AI. You can use training pipelines to create an AutoML-trained model or a custom-trained model. For custom-trained models, training pipelines orchestrate custom training jobs and hyperparameter tuning with additional steps like adding a data set or uploading the model to Vertex AI for prediction serving. Learn more

Region: us-central1 (Iowa)

Name	ID	Status	Job type	Model type	Created	Elapsed time	Labels
PCM_data	6419078900206272512	✓ Finished	Training pipeline	Text classification (Single-label)	21 Jul 2022, 22:19:10	8 hr 14 min	–

Model



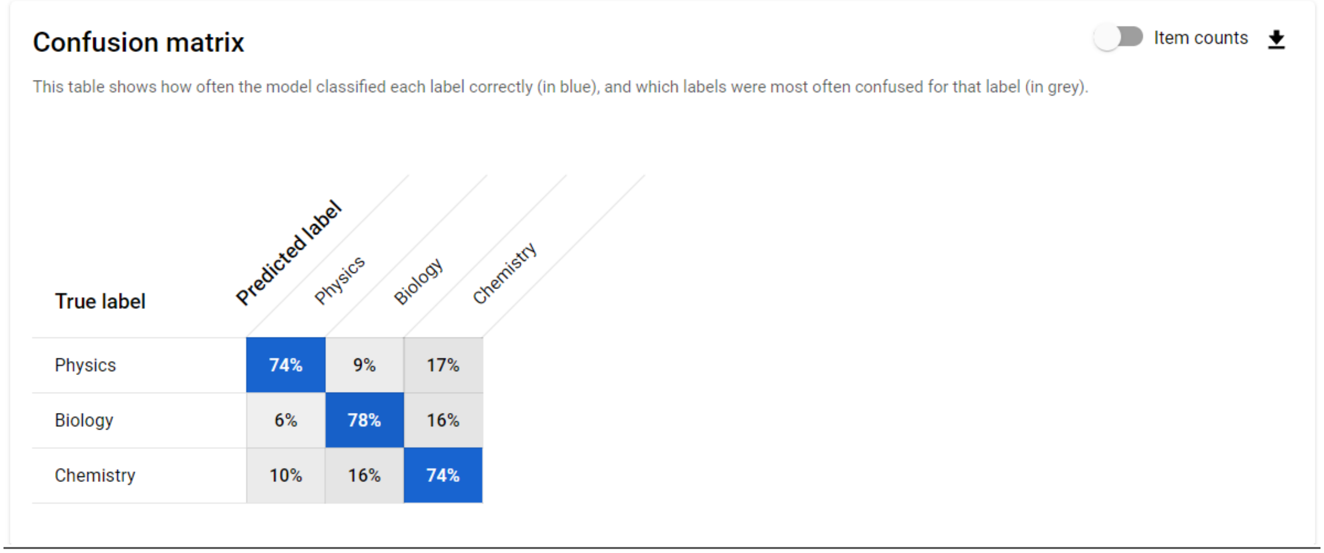
Confidence threshold 0.5

All labels	0
Biology	0.89065
Physics	0.84203
Chemistry	0.81729

All labels

Average precision	0.854
Precision	81.6%
Recall	65.2%
Created	22 Jul 2022, 06:33:16
Total items	7,947
Training items	6,357
Validation items	795
Test items	795

To evaluate your model, set the **confidence threshold** to see how precision and recall are affected. The best confidence threshold depends on your use case. Read some example scenarios to learn how evaluation metrics can be used.

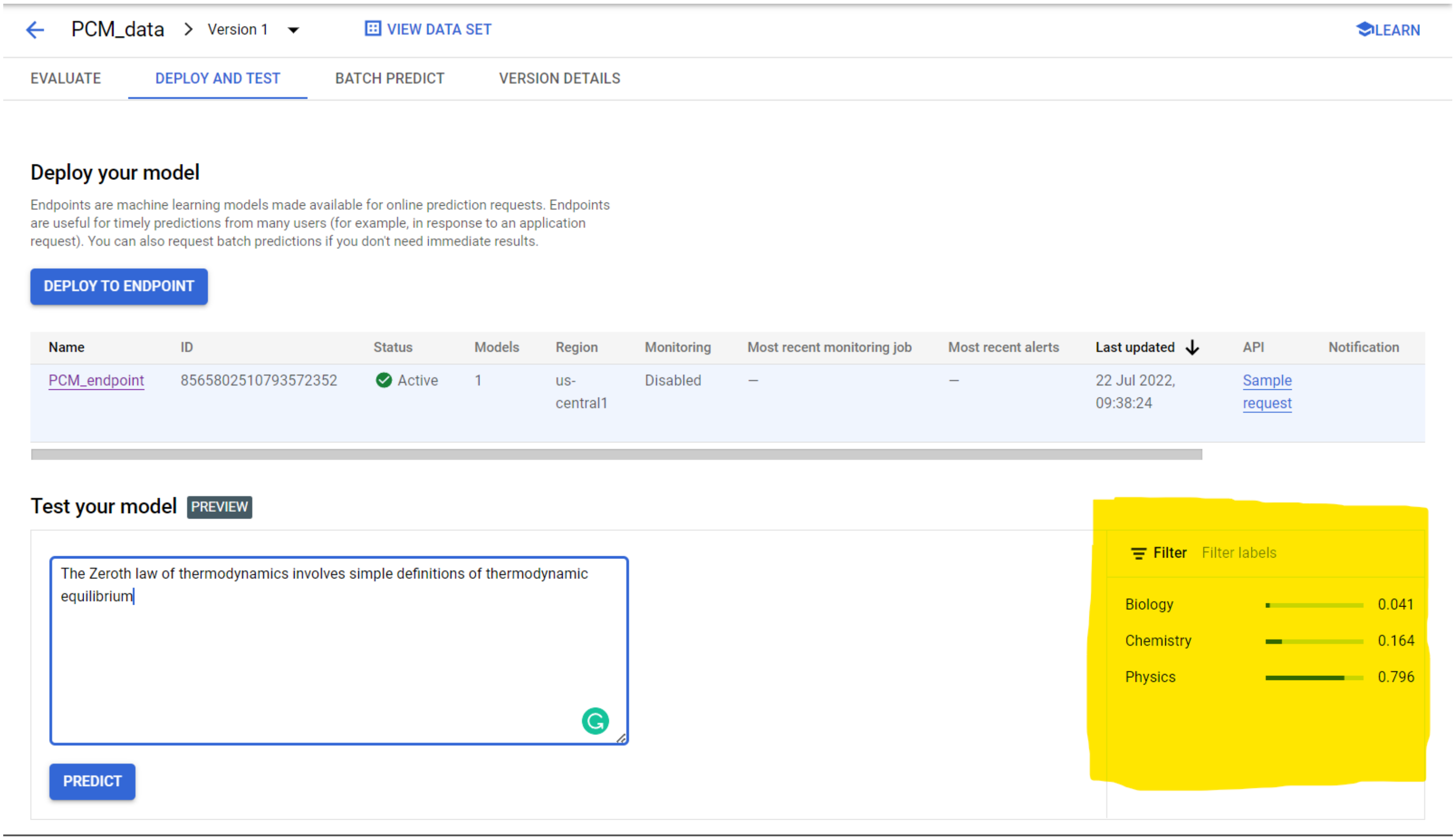


Confusion matrix

This table shows how often the model classified each label correctly (in blue), and which labels were most often confused for that label (in grey).

True label \ Predicted label	Physics	Biology	Chemistry
Physics	74%	9%	17%
Biology	6%	78%	16%
Chemistry	10%	16%	74%

Classifying The text



Deploy your model

Endpoints are machine learning models made available for online prediction requests. Endpoints are useful for timely predictions from many users (for example, in response to an application request). You can also request batch predictions if you don't need immediate results.

Name	ID	Status	Models	Region	Monitoring	Most recent monitoring job	Most recent alerts	Last updated ↓	API	Notification
PCM_endpoint	8565802510793572352	✓ Active	1	us-central1	Disabled	–	–	22 Jul 2022, 09:38:24	Sample request	

Test your model

The Zeroth law of thermodynamics involves simple definitions of thermodynamic equilibrium

Biology	0.041
Chemistry	0.164
Physics	0.796