[INFERENCE]
model4: code/requirements.txt X
model5: code/requirements.txt + code O
inference.py format: https://www.gravitywell.co.uk/insights/serving-inferences-from-your-machine-learning-model-with-sagemaker-and-tensorflow/

✓ Locations of extra files

✓ Importing modules (esp gluonnlp)***WORKS in Jupyter for some reason?
>> Tried to do the same thing in Jupyter (only error: JSON input format)
>> Commented out .read(): is not JSON serializable
>> Only added .decode():
WHAT TYPE OF OBJECT IS DATA?! -> 'gunicorn.http.body.Body'

req.json = json.loads(body.decode('utf-8')) NO

request.stream.read() NO
>> json.loads(body.decode('utf-8')) NO

json.loads(data): the JSON object must be str, bytes or bytearray, not 'Body'
OR request.post(host, json=data)

>> import os (was commented out-is this the problem? what)

<u>CHECK: requirements (modules), input, output</u>

✓ Tests-DEFAULT inference.py: correct input format for requests.post
>> TEST1: decoded_data = data.read().decode('utf-8')
{'error': 'JSON Value: "{\\"instances\\"": [[2, 529, 5944, 7176, 7673, 6579, 6043, 606, 423, 420, 45, 6141, 6682, 2152, 6855, 7443, 4285, 7227, 3]]}" Is not object'}
-> oops

>> TEST2: decoded_data = data.read()
"{"error": "Object of type 'bytes' is not JSON serializable"}"
-> happened at json.dumps()

>> TEST3: decoded_data = data.read() AND ...data=json.loads(decoded_data)
{'error': 'JSON Parse error: Invalid value. at offset: 0'}
-> apparently the solution to this is to: data=json.dumps(whatever the input is)

>> TEST4: decoded_data = json.loads(data.read()) AND ...
data=json.dumps(decoded_data): SUCCESS!

```
import requests
import json
import os

def handler(data, context):
    decoded_data = json.loads(data.read())
    response = requests.post(context.rest_uri, data=json.dumps(decoded_data))
    response_content_type = context.accept_header
    return response.content, response_content_type
```

CHECK: input format-for now, STRING product name

○ Tests-from DEFAULT code to actual io processing code (inference.py)

>> TEST1: import to inference.py-other modules (files and ones installed using pip)

- ~~Should comment out ‘import sys’~~
- Function ‘makeUNKTokenList’ included in the code

```
File "/opt/ml/model/code/inference.py", line 6, in <module>
    import torch
ModuleNotFoundError: No module named 'torch'
```

>> probably because requirements.txt in notebook cannot be used
(should be in tar.gz model archive)

1) py_version=‘py3’: unexpected keyword argument
-> X (tried this because NEED PYTHON3 (pip3 install torch) to install torch)

2) Put only requirements in tar.gz/code + dependencies=‘requirements.txt’
-> X

```
WARNING:__main__:loading modules in "/opt/ml/model/code/lib", ignoring requirements.txt
```

(torch not found again)

3) Still have requirements in notebook code directory + dependencies='code/requirements.txt'
-> X

>> following error from 2)
4) Follow https://github.com/aws/sagemaker-python-sdk/issues/1071 and specify source directory
-> UGH

```
src/gluonnlp/data/fast_bert_tokenizer.c:4:10: fatal error: Python.h: No such file or directory
     #include "Python.h"
              ^~~~~~~~~~
    compilation terminated.
    error: command 'x86_64-linux-gnu-gcc' failed with exit status 1

ModuleNotFoundError: No module named 'torch'
```

5) Include lxml in requirements.txt
-> NO

6) framework_version='1.13'
-> UH…NO

☑ Correct parameters (content_type='application/json')

☑ Response object error (see Test B-guessing this is because the returned output is not in json format)

Refer to https://github.com/aws/sagemaker-tensorflow-serving-container/pull/19/files

☑ {'error': 'JSON Parse error: Invalid value. at offset: 0'}

- json.dumps(data) EZ

>> DEFAULT inference.py WORKS! >> ~~gluonnlp is the only problem~~

☑ Input text file for local inference code (output text file)

A. Without extra files and gluonnlp + DEFAULT inference.py: parameter error
B. ^ with two values returned : ModelError ('Response' object has no attribute 'encode')

- 'Response' object = from requests.post (1. Not subscriptable)

Lambda: text file conversion
-> inference.py: uses text file, split products by line -> code/ decode (IMPLEMENTED)

[BATCH TRANSFORM]

☑ Complete batch transform using single array JSON files (2)

- JSON format (" for keys, not ')

>> When io handling is possible:

○ Perform batch transform to Korean text files?

[DOCKER] local

- Test out image on Docker (is it working): is NOT local mode!

>> get model
And serve it >> then where should I put inference.py (it could just invoke an endpoint)
requirements.txt would be it's own thing, and I would install modules from it in the container
BUT THEN inference.py should be with the model
Does tensorflow/serving recognize this??? (I thought this applied to SageMaker ONLY)

- Endpoint (using full requirements.txt and inference.py): torch is now a problem?

User name	Password	Access key ID	Secret access key	Console login link
saeyoon.kim1@cj.net	1q2w3e4r!	AKIA5SIZZPEDDP5NHPWG	lO4YHN7XKG4He675OzI1qZfPpclgUg5SeC9cUB7T	https://dk.signin.aws.amazon.com/console