Capstone Project Weekly Progress Report

Project Title	MEALBUDDY
Group Name	GROUP_G
Student names/Student IDs	ELVIN IYPE MATHEW C0769974 ELDA VARGHESE C0769741 TOM JOSEPH C0760915 JEENA HELEN FRANCIS C0764493 CHINJU BABY C0769912
Reporting Week	02 AUGUST 2020 - 08 AUGUST 2020
Faculty Supervisor	WILLIAM POURMAJIDI

1. Tasks Outlined in Previous Weekly Progress Report

- **Continue with final report**
- **Continue with Presentation**
- **Deploy Endpoint for XGBosst using sagemaker**
- **Demo test run**
- **Test scenarios**

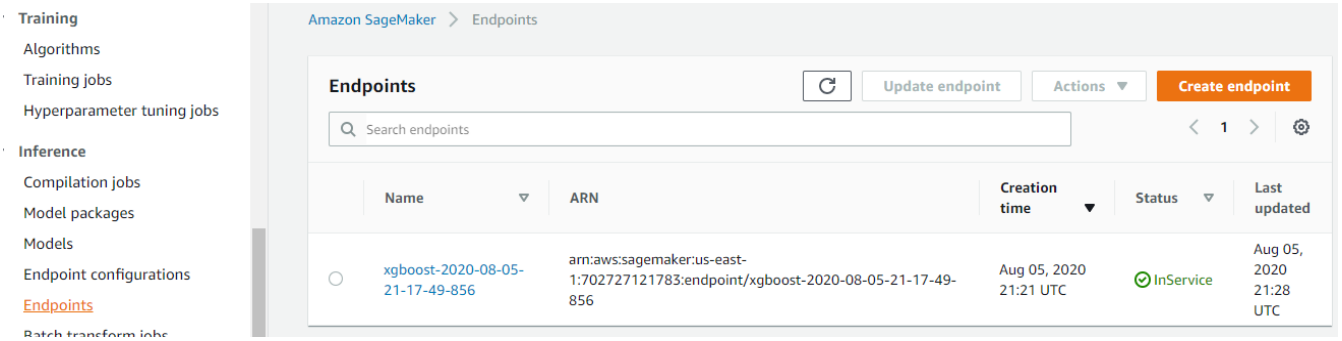
2. Progress Made in Reporting Week

- **The progress on the final report is going well as well as along with the final presentation. The different weekly reports are revisited for information/points to include in the report and presentation.**
- **To deploy Endpoint for XGBosst using sagemaker**
- **We used the XGBoost built-in algorithm to build an XGBoost training container. Spot the XGBoost built-in algorithm image URI using the Amazon SageMaker get_image_uri API.**
- **After specifying the XGBoost image URI, you can use the XGBoost container to construct an estimator using the SageMaker Estimator API and initiate a training job. This XGBoost built-in algorithm mode does not incorporate your own XGBoost training script and runs directly on the input datasets.**

- **After you fit an XGBoost Estimator, you can host the newly created model in SageMaker.**
- **After you call fit, you can call deploy on an XGBoost estimator to create a SageMaker endpoint. The endpoint runs a SageMaker-provided XGBoost model server and hosts the model produced by your training script, which was run when you called fit. This was the model you saved to model_dir.**

```
[82]: from sklearn.metrics import accuracy_score
      accuracy_score(val_label, pred_val)

[82]: 1.0

[108]: xgb_predictor.endpoint

[108]: 'xgboost-2020-08-05-21-17-49-856'

[109]: #sagemaker.Session().delete_endpoint(xgb_predictor.endpoint)
       #bucket_to_delete = boto3.resource('s3').Bucket(bucket)
       #bucket_to_delete.objects.all().delete()
```

- **Deploy returns a Predictor object, which you can use to do inference on the Endpoint hosting your XGBoost model. Each Predictor provides a predict method which can do inference with numpy arrays, Python lists, or strings. After inference arrays or lists are serialized and sent to the XGBoost model server, predict returns the result of inference against your model.**
-



- **The demo test runs are being tested and trying to record it.**

```
#call dynamodb search
searchResponse = searchYelpRestaurant.search(location,cuisine,dining_date,dining_time,num_people,phone)

#make prediction
predictionList = callAIModel.makePrediction(searchResponse)

restuarantList = []
for item in responseList:
    restuarantList.append(item.split(',')[1]+" call @"+item.split(',')[2])

sendMessage = "Enjoy "+", ".join(restuarantList)
if(len(sendMessage)>15):
    logger.debug("More than 160 characters "+str(len(sendMessage)))

#sendMessage = sendSNS.sendMessageNotification(phone,sendMessage)
```

lambda_function.py
saveUserRequest.py
searchYelpRestaurant.py
sendSNS.py

```python

def recommendSorter(e):
  return e[0]

def makePrediction(payload):

    responseList = []

    endpoint_name = 'xgboost-2020-08-05-21-17-49-856'
    runtime = boto3.Session().client(service_name='sagemaker-runtime',region_name='us-east-1')

    #review_count   rating  cuisine_american    cuisine_chinese cuisine_greek
    #cuisine_indian cuisine_italian cuisine_latin   cuisine_mexican cuisine_persian cuisine_spanish

    #125.0,3.5,0.0,0.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0

    for item in payload:
        inputData = ",".join(item[2:4])
        if(item[4] == 'american'):
            inputData = inputData + ",1,0,0,0,0,0,0,0"
        elif(item[4] == 'chinese'):
            inputData = inputData + ",0,1,0,0,0,0,0,0"
        elif(item[4] == 'greek'):
            inputData = inputData + ",0,0,1,0,0,0,0,0"
        elif(item[4] == 'indian'):
            inputData = inputData + ",0,0,0,1,0,0,0,0"
        elif(item[4] == 'italian'):
            inputData = inputData + ",0,0,0,0,1,0,0,0"
        elif(item[4] == 'latin'):
            inputData = inputData + ",0,0,0,0,0,1,0,0"
        elif(item[4] == 'spanish'):
            inputData = inputData + ",0,0,0,0,0,0,1,0"
        elif(item[4] == 'spanish'):
            inputData = inputData + ",0,0,0,0,0,0,0,1"

        try:
            response = runtime.invoke_endpoint(EndpointName=endpoint_name, ContentType='text/csv', Body=inputData)
            recommend = float(response['Body'].read())
            responseList.append(str(recommend)+","+','.join(item))
        except Exception as ex:
            raise ex

    responseList.sort(key=recommendSorter)

    itr = 0
    topResponseList = []
    for item in responseList:
        topResponseList.append(item)
        itr = itr+1
        if(itr>3):
            break

    return topResponseList
```

1:1 Pyth

3. Difficulties Encountered in Reporting Week

- **While coding using the sagemaker in a jupyter notebook instance. we had two difficulties**
  - **Permission to allow full access for lambda role to execute sagemaker. We fixed it by adding the required policy to the lambda role.**

```
Function logs:
START RequestId: d9b3c55b-a42e-4b19-a7e8-a323c65355b8 Version: $LATEST
98,4.5,0,0,0,1,0,0,0,0,0
[ERROR] ClientError: An error occurred (AccessDeniedException) when calling the InvokeEndpoint operation: User: arn:aws:sts::702727121783:ass
Traceback (most recent call last):
  File "/var/task/lambda_function.py", line 13, in lambda_handler
    response = runtime.invoke_endpoint(EndpointName=endpoint_name, ContentType='application/x-image', Body=payload)
  File "/var/runtime/botocore/client.py", line 316, in _api_call
    return self._make_api_call(operation_name, kwargs)
  File "/var/runtime/botocore/client.py", line 635, in _make_api_call
    raise error_class(parsed_response, operation_name)END RequestId: d9b3c55b-a42e-4b19-a7e8-a323c65355b8
REPORT RequestId: d9b3c55b-a42e-4b19-a7e8-a323c65355b8  Duration: 628.65 ms Billed Duration: 700 ms Memory Size: 128 MB Max Memory Used: 77 M
```

  - **The data format for training and validation for the machine model. It requires the output label to be first in column format.**

```
Parameter image_name will be renamed to image_uri in SageMaker Python SDK v2.
2020-08-05 03:26:59 Starting - Starting the training job...
2020-08-05 03:27:01 Starting - Launching requested ML instances.........
2020-08-05 03:28:43 Starting - Preparing the instances for training......
2020-08-05 03:29:55 Downloading - Downloading input data.........
2020-08-05 03:31:27 Training - Training image download completed. Training in progress..Arguments: train
[2020-08-05:03:31:27:INFO] Running standalone xgboost training.
[2020-08-05:03:31:27:INFO] File size need to be processed in the node: 0.02mb. Available memory size in the node: 8484.77mb
[2020-08-05:03:31:27:INFO] Determined delimiter of CSV input is ','
[03:31:27] S3DistributionType set as FullyReplicated
[03:31:27] 2319x1 matrix with 2319 entries loaded from /opt/ml/input/data/train?format=csv&label_column=0&delimiter=,
[2020-08-05:03:31:27:INFO] Determined delimiter of CSV input is ','
[03:31:27] S3DistributionType set as FullyReplicated
[03:31:27] 580x1 matrix with 580 entries loaded from /opt/ml/input/data/validation?format=csv&label_column=0&delimiter=,
[2020-08-05:03:31:27:ERROR] Customer Error: Label must be in [0,1] for logistic regression task. If input is in csv format,
ensure the first column is the label.
Traceback (most recent call last):
  File "/opt/amazon/lib/python2.7/site-packages/sage_xgboost/train.py", line 41, in main
    standalone_train(resource_config, train_config, data_config)
  File "/opt/amazon/lib/python2.7/site-packages/sage_xgboost/train_methods.py", line 19, in standalone_train
    train_job(resource_config, train_config, data_config)
  File "/opt/amazon/lib/python2.7/site-packages/sage_xgboost/train_helper.py", line 341, in train_job
    "Label must be in [0,1] for logistic regression task. If input is in csv format, ensure the first "
CustomerError: Label must be in [0,1] for logistic regression task. If input is in csv format, ensure the first column is th
e label.

2020-08-05 03:31:32 Uploading - Uploading generated training model
```

- **The data format for endpoint execution. The input parameters length and format should match the invocation endpoint function.**

```
Function logs:
START RequestId: 37cafa54-08e2-4a3a-a75e-11f417481c79 Version: $LATEST
98,4.5,0,0,0,1,0,0,0,0,0
[ERROR] ModelError: An error occurred (ModelError) when calling the InvokeEndpoint operation: Received client error (415) from model with mes
Traceback (most recent call last):
··File "/var/task/lambda_function.py", line 13, in lambda_handler
····response = runtime.invoke_endpoint(EndpointName=endpoint_name, ContentType='application/x-image', Body=payload)
··File "/var/runtime/botocore/client.py", line 316, in _api_call
····return self._make_api_call(operation_name, kwargs)
··File "/var/runtime/botocore/client.py", line 635, in _make_api_call
····raise error_class(parsed_response, operation_name)END RequestId: 37cafa54-08e2-4a3a-a75e-11f417481c79
REPORT RequestId: 37cafa54-08e2-4a3a-a75e-11f417481c79  Duration: 634.02 ms Billed Duration: 700 ms Memory Size: 128 MB Max Memory Used: 78 M
```

4. Tasks to Be Completed in Next Week

- **Finish final report**
- **Finish Presentation**