Amazon Inventory Reconciliation Using AI

AWS Machine Learning Engineer Nanodegree

1. Domain Background:

The emerging technology advancements force retailers to assess their current business approaches and systems, especially warehouse sophistication. As a key element of the entire supply chain, the main goal of the warehouse is to provide maximum effectiveness, along with productivity optimization, for satisfying the requirements of the omnichannel retailer.

But, according to Gartner, "60% of companies are dissatisfied with the fit between their supply chain planning objectives and supply planning capability." Today, clients prefer accustomed full-cycle product offerings, which increases their loyalty to retailers that can offer the necessary products at any time. Therefore, companies must refine their warehouses to address the supply chain challenges.

To achieve organic expansion and growth, both retailers and warehouse managers need to assess the current state of the warehouse, uncover its potential challenges, and then build upon its major strengths. At the same time, managers should apply the right practices for providing accurate data, insights, or trends and diagnosing and resolving issues.

2. Problem Statement:

Inventory management is critical to Amazon's success. Thus, the need arises to apply artificial intelligence to assure the correctness of deliveries.

Amazon Fulfillment Centers are bustling hubs of innovation that allow Amazon to deliver millions of products to over 100 countries worldwide. These products are randomly placed in bins, which are carried by robots.

Occasionally, items are misplaced while being handled, resulting in a mismatch: the recorded bin inventory, versus its actual content.

The project predicts the number of items in a bin, thus detecting any inventory variance. By correcting variance upon detection, Amazon will better serve its customers.will better serve its customers.

3. Datasets and inputs:

Amazon has made public the Bin Image Dataset. It contains images and metadata from bins of a pod in an operating Amazon Fulfillment Center. The bin images in this dataset are captured as robot units carrying pods as part of normal operations. The Bin Image dataset provides the metadata for each image from where the number of items in the bin can be derived.

4. Solution Statement:

Using a pre-trained model (Resnet 50) for feature extractions from our images data by freezing all the weights of the neural network and only changing the classification layer so the model needs to train these weights at the last classification layer, using Stochastic Gradient Descent with categorical cross-entropy loss function for 5 classes only.

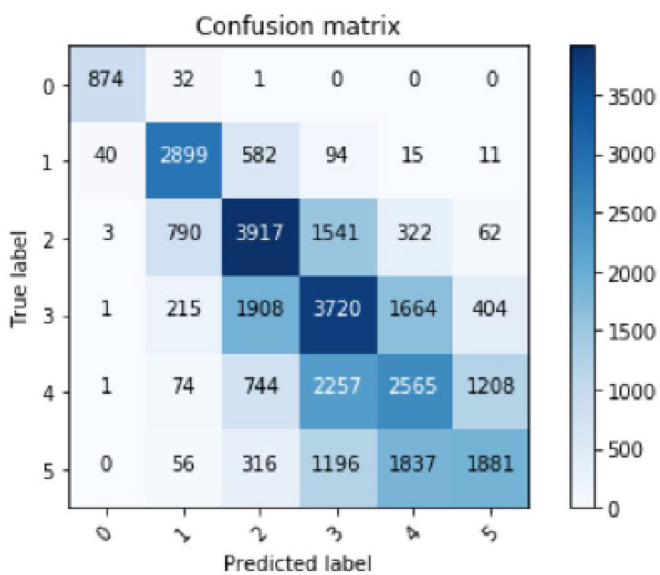
- The Algorithm will be built using Resnet 50 pre-trained model.
- Deep Learning Framework – PyTorch.
- Implement Adam optimizer to train the model.
- A corresponding Sage Maker instance will be created and data will be fed from the S3 bucket.
- The model will also be tuned to find out the best hyper-parameters.

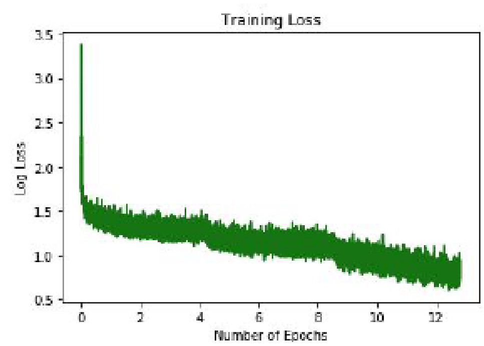
5. A benchmark model:

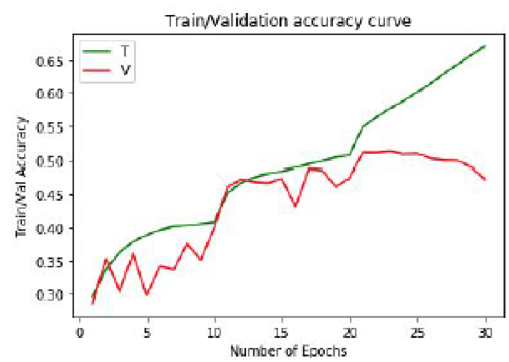
Refer to Stanford University Palo Alto, California results:

The project's repository is: https://github.com/OneNow/AIInventory-Reconciliation

Model	Epochs	Train Accuracy	Test Accuracy	Test Root Mean Square Error
ResNet 18 (SGD)	20	55.9	50.4	0.98
ResNet 34 (SGD)	22	55.2	51.2	0.99
ResNet 34 (SGDR)	36	57.8	53.8	0.94



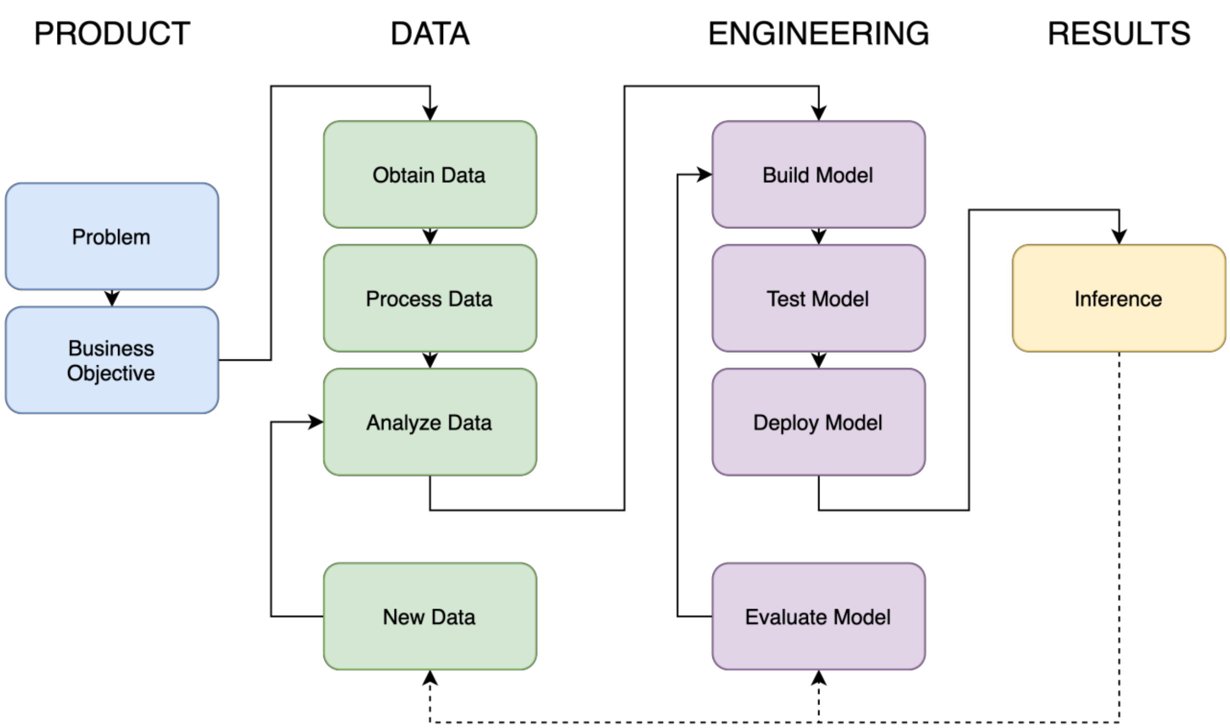




6. Evaluation Metrics:

Since it is a classification problem, the overall accuracy of the classification and F1 score can be used to evaluate the performance of the trained model.

7. Workflow:



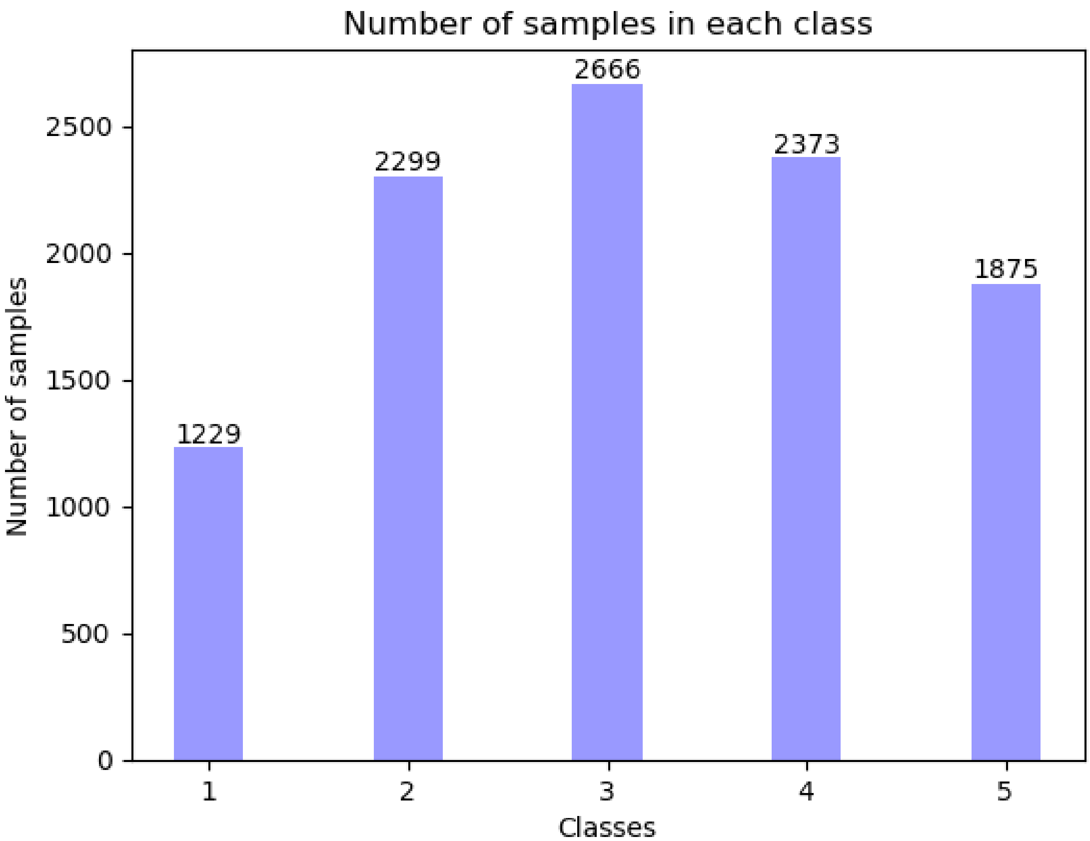
Data:

These are some typical images in the dataset. A bin contains multiple object categories and a various number of instances. The corresponding metadata exists for each bin image and it includes the object category identification (Amazon Standard Identification Number, ASIN), quantity, size of objects, weights, and so on. The size of bins varies depending on the size of objects in them. The tapes in front of the bins are for preventing the items from falling out of the bins and sometimes it might make the objects unclear. Objects are sometimes heavily occluded by other objects or limited viewpoints of the images.

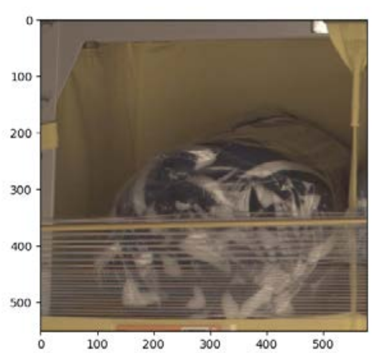


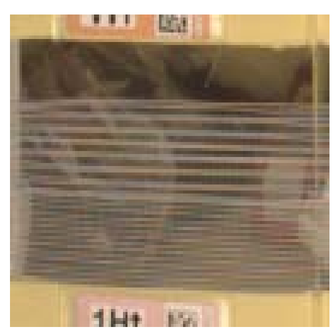
I used a part of the data that:

- Consists of 10000 Images of different sizes
- Restricted project to only images with fewer than 6 bin items
- Histogram of distribution mass, for the number of objects in a bin:



- Some images are occluded
- Even for a human, some images are difficult to classify





Data extraction:

```
  0%|          | 1/1228 [00:00<02:48,  7.28it/s]
Downloading Images with 1 objects
100%|██████████| 1228/1228 [01:45<00:00, 11.59it/s]
  0%|          | 1/2299 [00:00<03:57,  9.67it/s]
Downloading Images with 2 objects
100%|██████████| 2299/2299 [03:23<00:00, 11.30it/s]
  0%|          | 2/2666 [00:00<03:23, 13.08it/s]
Downloading Images with 3 objects
100%|██████████| 2666/2666 [03:58<00:00, 11.19it/s]
  0%|          | 2/2373 [00:00<03:01, 13.09it/s]
Downloading Images with 4 objects
100%|██████████| 2373/2373 [03:26<00:00, 11.48it/s]
  0%|          | 2/1875 [00:00<02:04, 15.08it/s]
Downloading Images with 5 objects
100%|██████████| 1875/1875 [02:39<00:00, 11.77it/s]
```

Data preprocessing:

Before model training, images were normalized:

- Re-sized to 224x224 pixels
- the dataset was augmented with horizontal flips
- Using random cropping

Hyperparameter tuning:

The following hyper-parameters are identified for tunning.
1. Learning Rate
2. Batch Size
3. Epoch

The Learning rate for the Adam optimizer is identified as one of the hyper-parameters which could help improve the training process rather than the predefined default.
Batch Size could help us optimize the computation speed and convergence.
The Number of Epochs could help the model to better train the model over a long time and different possibilities can be checked for tunning.
For the experiment, 5 Jobs are created for the tunning and one among them is chosen based on the results i.e., objective metric.

	batch_size	epochs	learning_rate	TrainingJobName	TrainingJobStatus	FinalObjectiveValue	TrainingStartTime	TrainingEndTime	TrainingElapsedTimeSeconds
2	"64"	"36"	0.019463	pytorch-training-230219-1256-003-0c4c7457	Completed	49.0	2023-02-19 12:57:48+00:00	2023-02-19 13:16:59+00:00	1151.0
4	"32"	"36"	0.044768	pytorch-training-230219-1256-001-fdf1ee37	Completed	49.0	2023-02-19 12:57:45+00:00	2023-02-19 13:16:57+00:00	1152.0
1	"64"	"50"	0.007449	pytorch-training-230219-1256-004-fda008a6	Completed	47.0	2023-02-19 12:57:52+00:00	2023-02-19 13:17:03+00:00	1151.0
0	"128"	"50"	0.003267	pytorch-training-230219-1256-005-1332a900	Completed	46.0	2023-02-19 12:57:54+00:00	2023-02-19 13:17:05+00:00	1151.0
3	"32"	"50"	0.001070	pytorch-training-230219-1256-002-32efe621	Completed	46.0	2023-02-19 12:57:51+00:00	2023-02-19 13:16:33+00:00	1122.0

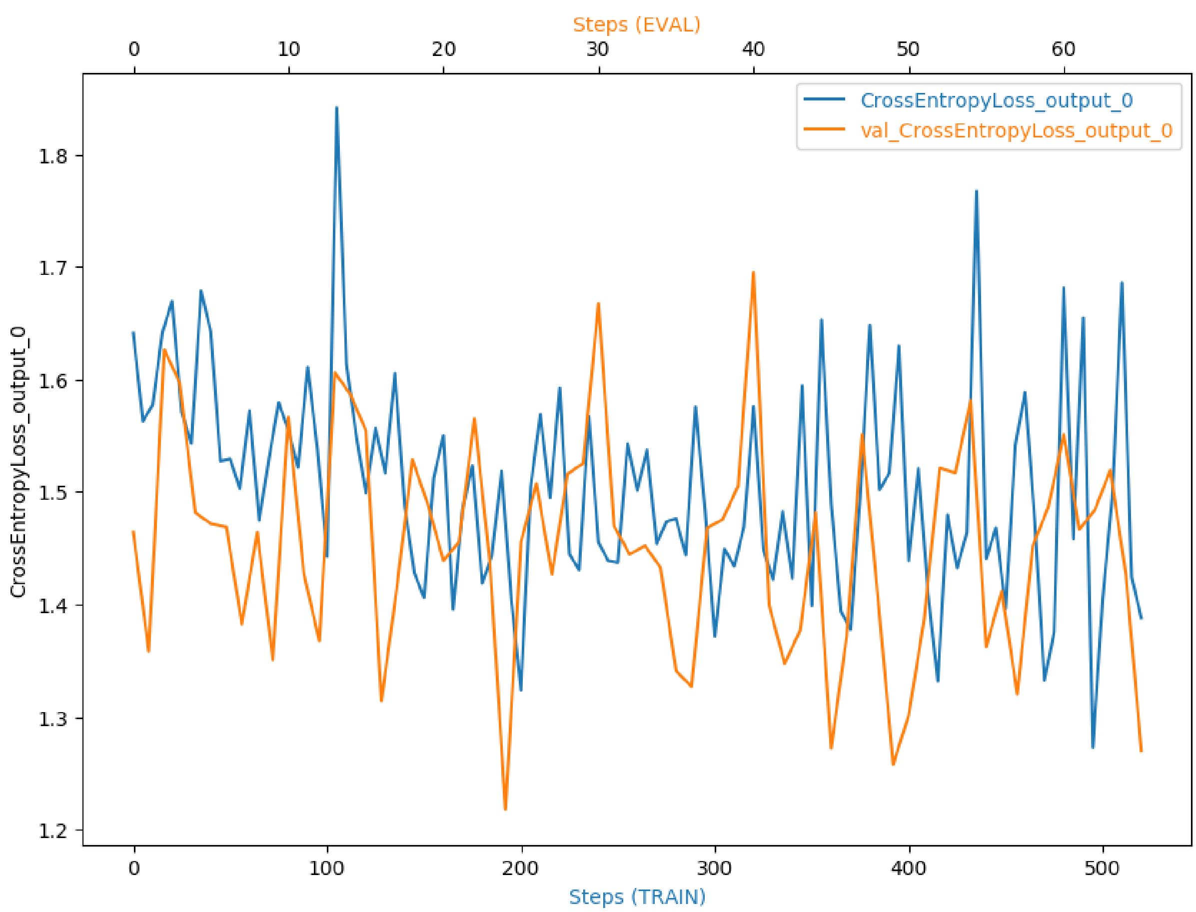
Hyper-parameter Tunning Jobs

Model:

I used Resnet 50 pre-trained model for feature extractions from our images data by freezing all the weights of the neural network and only changing the classification layer so the model only needs to train these weights at the last classification layer, I used a learning rate of 0.001, batch size 32, with 50 epochs and the loss function was categorical cross entropy for 5 classes.

Model Evaluation:

The accuracy of the benchmark model chosen is 53.8 %, the experiment model didn't achieve the results of the Benchmark because of it trained on a small part of the data so we can download the whole data to get more accuracy.



Cost Analysis:

1. The cost of cloud watch:

CloudWatch		$0.07
US East (N. Virginia)		**$0.07**
Amazon CloudWatch		$0.07
$0.00 per request - first 1,000,000 requests	640.000 Requests	$0.00
$0.01 per 1,000 metrics requested using GetMetricData API - US East (Northern Virginia)	37.000 Metrics	$0.00
$0.30 per metric-month for the first 10,000 metrics - US East (Northern Virginia)	0.217 Metrics	$0.07
AmazonCloudWatch PutLogEvents		$0.00
First 5GB per month of log data ingested is free.	0.047 GB	$0.00
AmazonCloudWatch USE1-TimedStorage-ByteHrs		$0.00
First 5GB-mo per month of logs storage is free.	0.001 GB-Mo	$0.00

2. The cost of sagemaker:

SageMaker	
US East (N. Virginia)	
Amazon SageMaker CreateVolume-Gp2	
$0.00 for SageMaker Debugger Built-in Rule Volume	126.025 GB-Mo
$0.14 per GB-Mo of Endpoint ML storage	0.007 GB-Mo
$0.14 per GB-Mo of Training Job ML storage	0.174 GB-Mo
Amazon SageMaker Invoke-Endpoint	
$0.016 per GB for Endpoint Data IN	0.000026 GB
$0.016 per GB for Endpoint Data OUT	0.000000250 GB
Amazon SageMaker RunInstance	
$0.0 for SageMaker Debugger Built-in Rule Instance	1.225 Hrs
$0.00 for SageMaker Debugger Built-in Rule Instance	2.976 Hrs
$0.05 per Studio-Notebook ml.t3.medium hour in US East (N. Virginia)	12.285 Hrs
$0.115 per Hosting ml.m5.large hour in US East (N. Virginia)	1.174 Hrs
$0.115 per Training ml.m5.large hour in US East (N. Virginia)	2.472 Hrs
$0.23 per Training ml.m5.xlarge hour in US East (N. Virginia)	1.419 Hrs

3. The cost of S3:

Simple Storage Service	
US East (N. Virginia)	
Amazon Simple Storage Service Requests-Tier1	
$0.005 per 1,000 PUT, COPY, POST, or LIST requests	121,272.000 Requests
Amazon Simple Storage Service Requests-Tier2	
$0.004 per 10,000 GET and all other requests	514,523.000 Requests
Amazon Simple Storage Service TimedStorage-ByteHrs	
$0.023 per GB - first 50 TB / month of storage used	0.026 GB-Mo

Total Cost:

Summary USD

Details + Expand All

AWS Service Charges	$2.34
CloudWatch	$0.07
Data Transfer	$0.00
Elastic File System	$0.04
Key Management Service	$0.03
SageMaker	$1.39
Simple Storage Service	$0.81

Future work:

- Remove the images from the training set where items are occluded
- Train the model on different training/ val / sets to reduce data distribution differences
- implement Adam optimizer to train CNN models
- Evaluate ensemble methods to improve the accuracy
- Use class weights to assign higher loss to classes with suboptimal performance

References:

https://github.com/silverbottlep/abid_challenge

https://view.publitas.com/p222-16173/amazoninventoryreconciliationusingai-pablorodriguezbertorello-nutchapoldendumrongsup-sravansripada/page/1