AI on IBM Z

Anti-money laundering solution template

This solution template provides an example on how to deploy AI using an IBM Z environment, while making use of open source frameworks, Machine Learning for IBM z/OS (MLz), and more.

Within this solution template, there are various phases of the AI lifecycle included. Work through each of the following steps to deploy your own anti-money laundering solution on IBM Z.

Table of contents

Step 1

AI model training

We will build an anti-money laundering AI model by training with the provided Rapid AI on IBM Z Development Jupyter notebook. Simply point the Jupyter notebook to your dataset and run it to generate your AI model. This trained AI model can then be deployed with MLz.

All sample code for this section is within

`aionz-st-anti-money-laundering/zST-model-training-jupyter`

Prerequisites

- Must have Python 3.9 & 3.11 installed

Dataset guidance

Sample anti-money laundering dataset can be found on Kaggle -

https://www.kaggle.com/datasets/ealtman2019/ibm-transactions-for-anti-money-laundering-aml

Required features

- Timestamp
- From Bank
- From Account
- To Bank
- To Account
- Amount Received
- Receiving Currency
- Amount Paid
- Payment Currency
- Payment Format
- Is Laundering

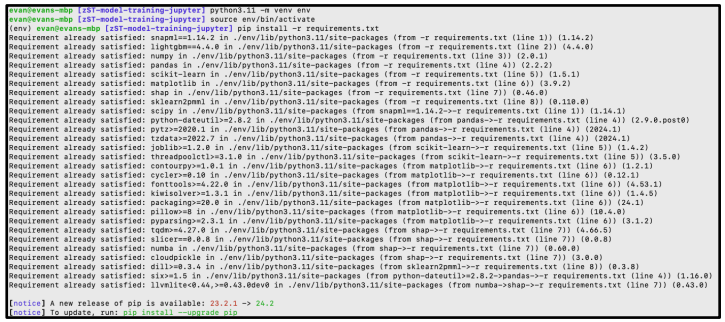
Configure rapid AI on IBM Z development environment

1. Create and activate Python virtual environment

   ```
   python3.9 -m venv env
   source env/bin/activate
   ```

2. Install required Python packages

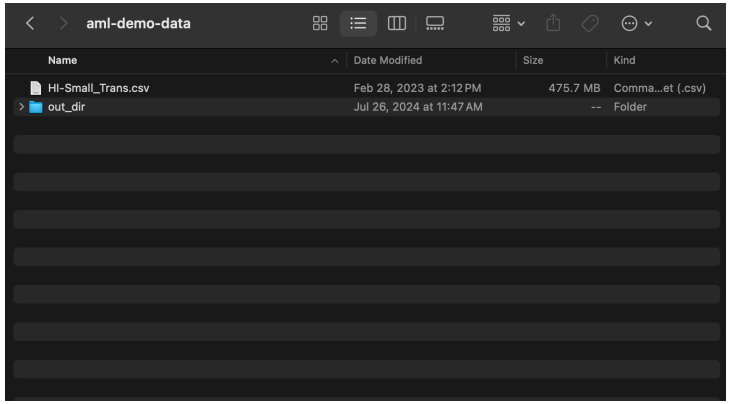
   ```
   pip install -r requirements.txt
   ```



Provide data

1. Put your input dataset (HI-Small_Trans.csv) in aml-demo-data/ directory

   Note: It is important to put the dataset here because it’s referenced by the data pre-processing and model training code (format_raw_transactions.py & aml_model_training.ipynb)



Data pre-processing

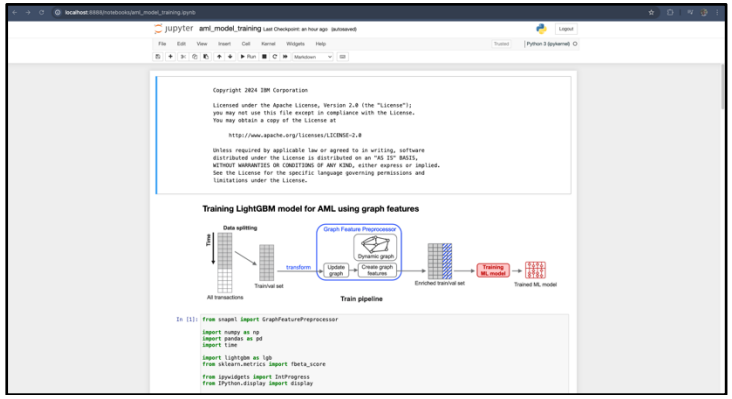
1. Run `python3.9 format_raw_transactions.py` to pre-process original dataset into a format suitable for model training (this will take several minutes to complete due to the volume of data, potentially up to 25 mins)

Model training

1. Run Jupyter (using python 3.11.5 kernel)

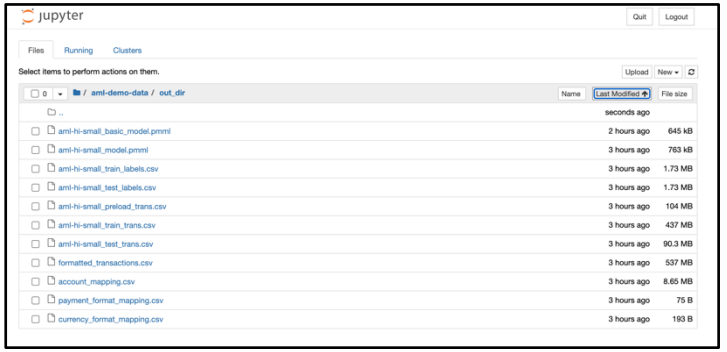
   `jupyter notebook`

2. View Jupyter interface
   a. Go to localhost:8888 in a web browser
3. Step through and run all cells within Jupyter notebook (aml_model_training.ipynb) within web browser



Access trained AI model

1. Once training is complete, you can find your AI models within the aml-demo-data/out_dir/ directory (aml_hi-small_model.pmml).



1. AI model training

2. AI model deployment

4. AI model integration

AI model training complete

Prerequisites

- Must have MLz 3.2 installed

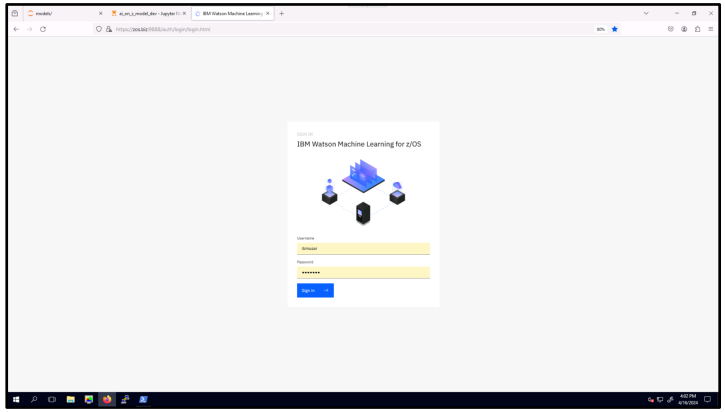
Step 2

AI model deployment

We will deploy our anti-money laundering AI model using MLz. We can utilize the model import functionality on the MLz UI. This deployed AI model can then be integrated into applications within the IBM Z environment.

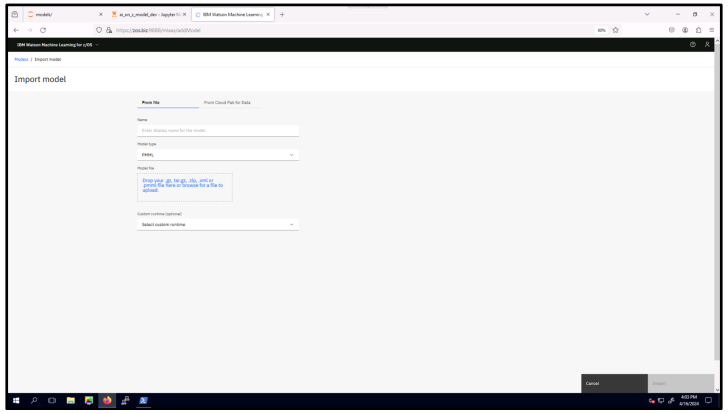
Go to MLz UI

1. Sign in with username/password

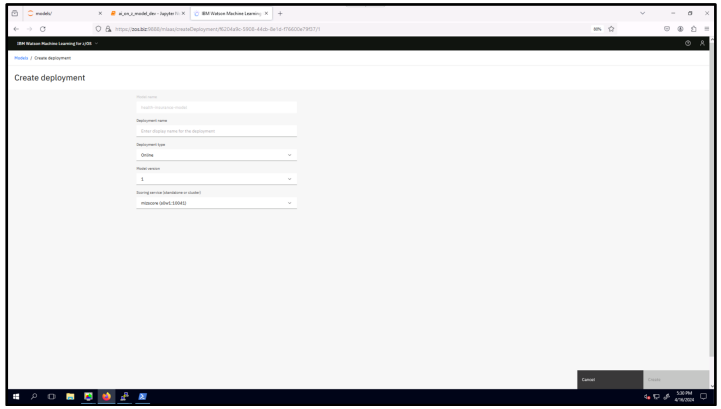


Import AI model

1. Go to models tab
2. Click import model
3. Enter model name
4. Choose model type
    a. Choose PMML if using your previously trained model
5. Drag and drop model file
   Use your previously trained model
6. Click import

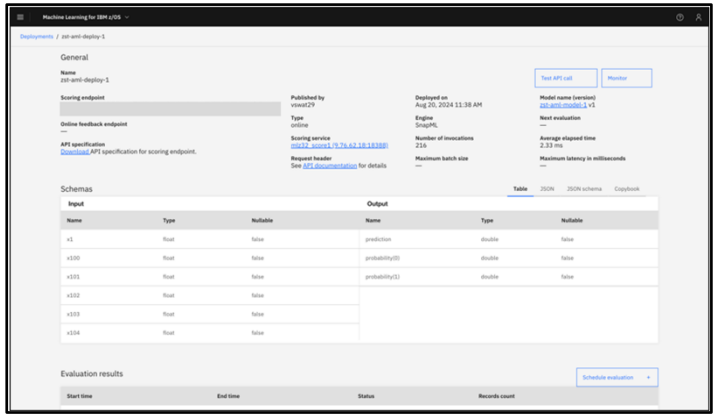


Deploy AI model



1. Go to models tab
2. Click action button for your model (on right side)
3. Click deploy
4. Enter deployment name
5. Choose deployment type
6. Choose model version
7. Choose scoring service
   Note: you should choose the correct scoring service based on your application (e.g. REST API)
8. Click create

View deployed AI model



1. Go to deployments tab
2. Click on action button for your deployed model (on right side)
3. Click view details

1. AI model training

2. AI model deployment

4. AI model integration

☑ AI model deployment complete

Step 3

AI model integration

We can use our deployed MLz anti-money laundering AI model and integrate it into different types of applications. Guidance on integrating the AI model into a sample anti-money laundering application is below.

All sample code for this section is within

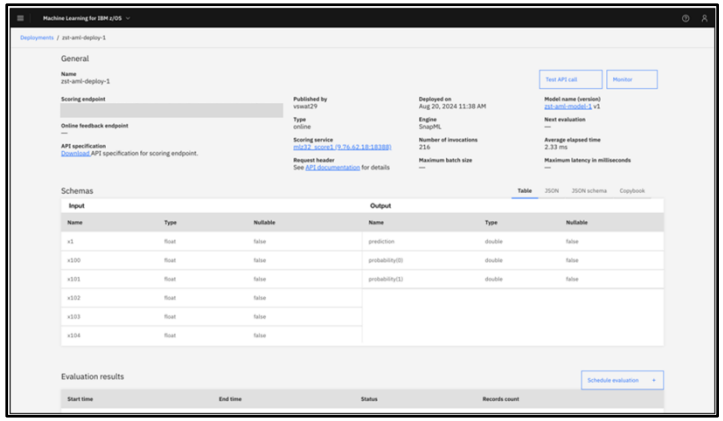
`aionz-st-anti-money-laundering/zST-model-integration-aml`

Prerequisites

- Must have node.js v16 or newer installed
- Must have Docker/podman installed
- Must have Git installed

Get model details for inferencing

1. Go to MLz UI
2. Go to deployments tab
3. Click on action button for your deployed model (on right side)
4. Click view details
5. Copy scoring endpoint

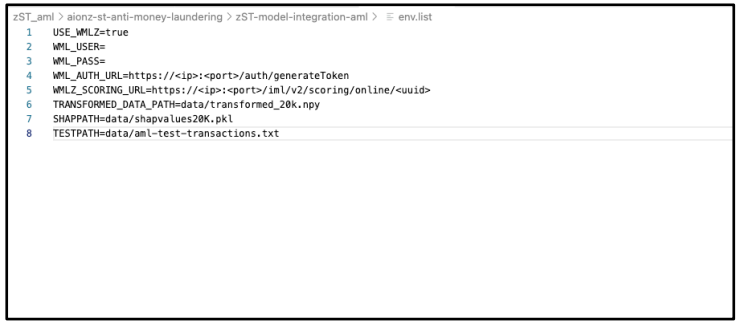


Configure sample application

1. Set the environment variables within

   `aionz-st-anti-money-laundering/zST-model-integration-aml/env.list`

   USE_WMLZ (set to true to ensure MLz is used)
   WML_USER (MLz username)
   WML_PASS (MLz password)
   WML_AUTH_URL (MLz auth token url)
   WMLZ_SCORING_URL (MLz scoring endpoint)



Build sample application

1. Run command in terminal

```
podman build -t aml .
```

Deploy sample application

1. Run command in terminal (e.g. port 9000)

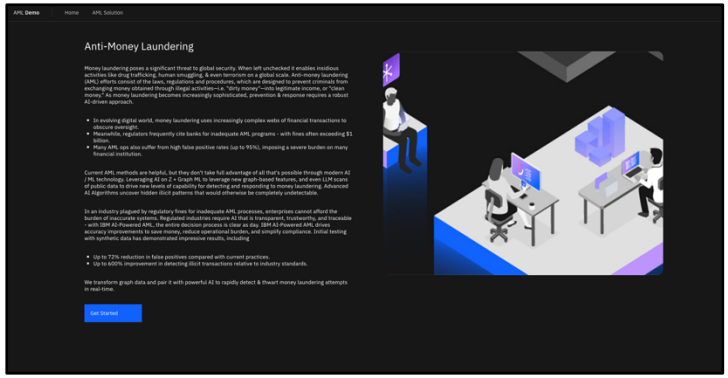
```
podman run -p 9000:80 --env-file env.list --
name aml-app aml
```

Access sample application

1. View the following URL in a web browser

   http://{ip address}:{port}/ui/

   ip address: IP of server you deployed application in
   port: port you used with podman run



☑ AI model integration complete