



AI on IBM Z

Data preprocessing solution template

This solution template provides an example on how to deploy AI with a preprocessing service using an IBM Z environment, while making use of open source frameworks, Machine Learning for 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 fraud detection solution using MLz on IBM Z.



Table of contents

AI model training.....	3
AI model deployment.....	8
Preprocessing pipeline deployment.....	12
AI model inferencing.....	16



Step 1

AI model training

We will build a fraud detection 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 TIS.

All sample code for this section is within

```
aionz-st-data-preprocessing-mlz/zST-model-training-jupyter
```

Prerequisites

- Must have Python (3.9 or 3.10) installed

Dataset guidance

Sample open source credit card transaction dataset can be found on Kaggle -

<https://www.kaggle.com/datasets/ealtman2019/credit-card-transactions>

There are several files included within the download. You can use credit_card_transactions_ibm_v2.csv for training. Due to the size of the sample dataset, the provided Jupyter notebook takes a subset of the data to decrease the training time. Please modify the code in the “Fetch and process data” cell of the provided Jupyter notebook later to use more data during training.

Required features

- User (integer) – unique ID for user making transaction
- Card (integer) – unique ID for credit card
- Year (integer) – year of the transaction
- Month (integer) – month of the transaction
- Day (integer) – day of the month of the transaction
- Time (integer) - time of the transaction (HH:MM)
- Amount (float) – dollar amount of the transaction
- Use Chip (string) – the type of transaction
- Merchant Name (integer) – unique ID for merchant name
- Zip (integer) – zip code of the transaction

1. AI model training

Access rapid AI on IBM Z development environment

Provide data

Model training

Access trained AI model

Access preprocessing pipeline

2. AI model deployment

3. Preprocessing pipeline deployment

4. AI model inferencing

Access rapid AI on IBM Z development environment

1. Access sample code

```
cd zST-model-training-jupyter
```

2. Create and activate Python virtual environment

```
python -m venv env source env/bin/activate
```

3. Install required Python packages

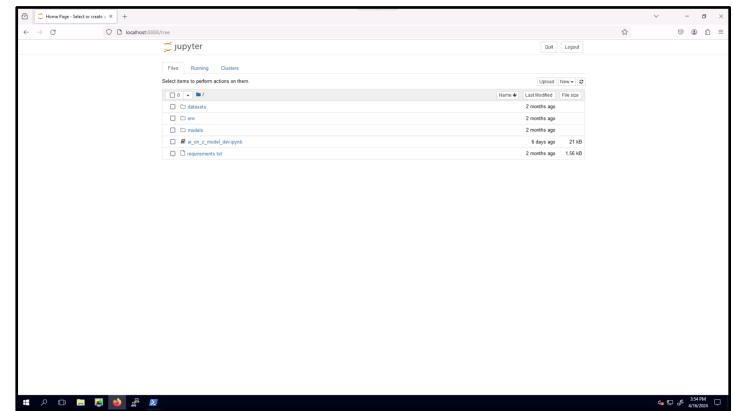
```
pip install -r requirements.txt
```

4. Run Jupyter

```
jupyter notebook
```

5. View Jupyter interface

- a. Go to localhost:8888 in a web browser



Access rapid AI on IBM Z development environment

Provide data

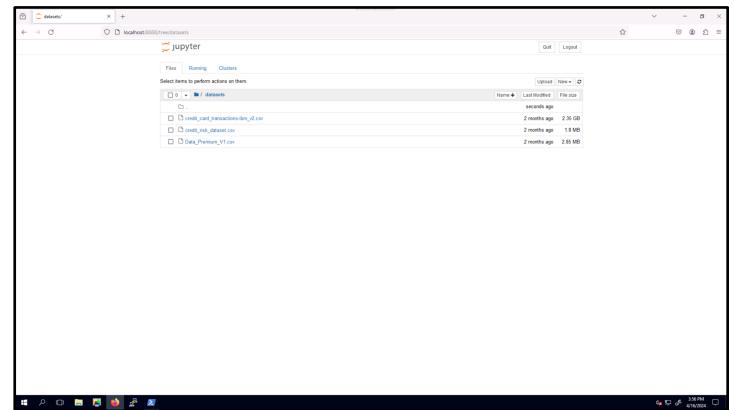
Model training

Access trained AI model

Access preprocessing pipeline

Provide data

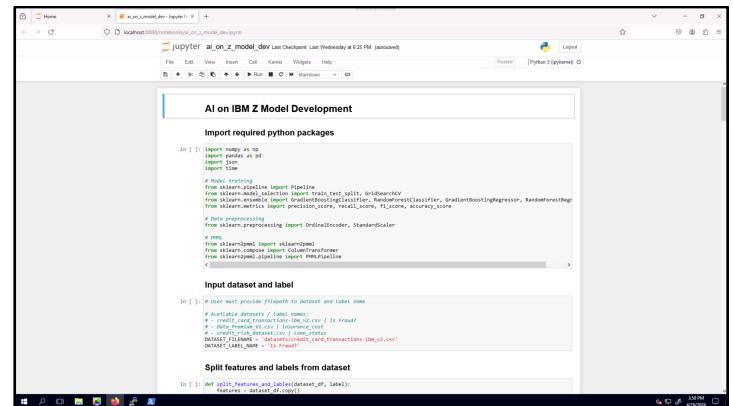
1. Your input dataset (csv) in `datasets/` directory
 2. Add input data to Jupyter notebook
`(ai_on_z_model_dev.ipynb)`
 - a. Set `DATASET_FILENAME` to the path to your dataset
 - b. Set `DATASET_LABEL_NAME` to the name of the column you're predicting from the dataset



Model training

1. Step through and run all cells within Jupyter notebook (`ai_on_z_model_dev.ipynb`) within web browser

Note: This may take several minutes



1. AI model training

Access rapid AI on IBM Z development environment

Provide data

Model training

[Access trained AI model](#)

[Access preprocessing pipeline](#)

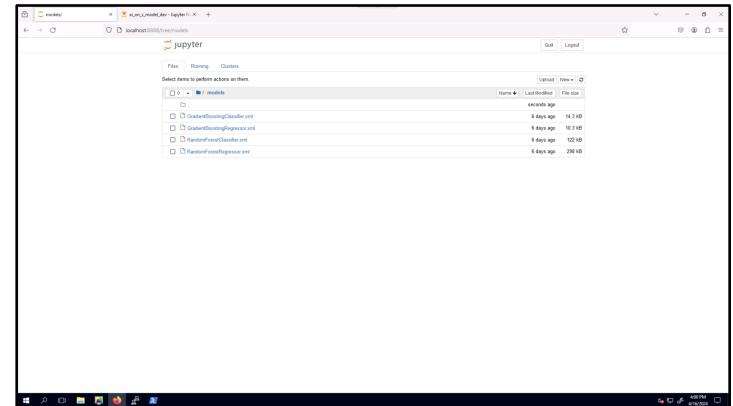
2. AI model deployment

3. Preprocessing pipeline deployment

4. AI model inferencing

Access trained AI model

- Once training is complete, you can find your AI models within the `models/` directory (choose one for the following AI model deployment step)



Access preprocessing pipeline

- Once training is complete, you can find your preprocessing pipeline within the `models/` directory (choose one for the following preprocessing deployment step)

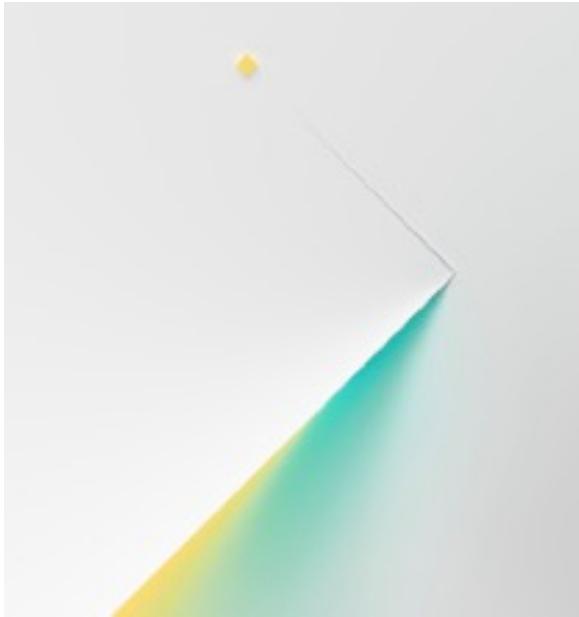
1. AI model training

2. AI model deployment

3. Preprocessing pipeline deployment

4. AI model inferencing

AI model training complete



Prerequisites

- Must have MLz installed

Step 2

AI model deployment

We will deploy our fraud detection 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.

1. AI model training

2. AI model deployment

3. Preprocessing pipeline deployment

4. AI model inferencing

[Go to MLz UI](#)

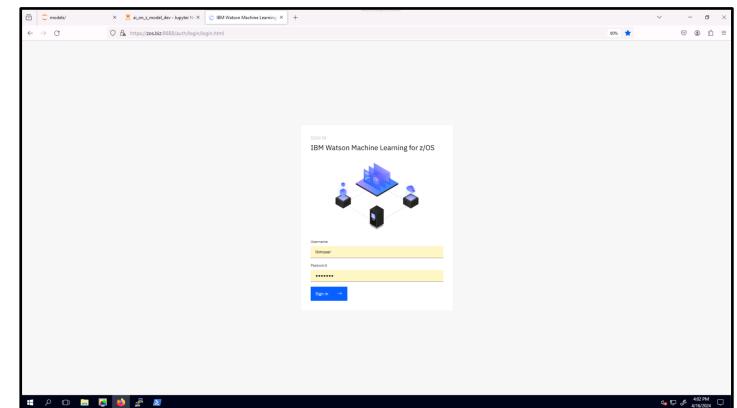
[Import AI model](#)

[Deploy AI model](#)

[View deployed AI model](#)

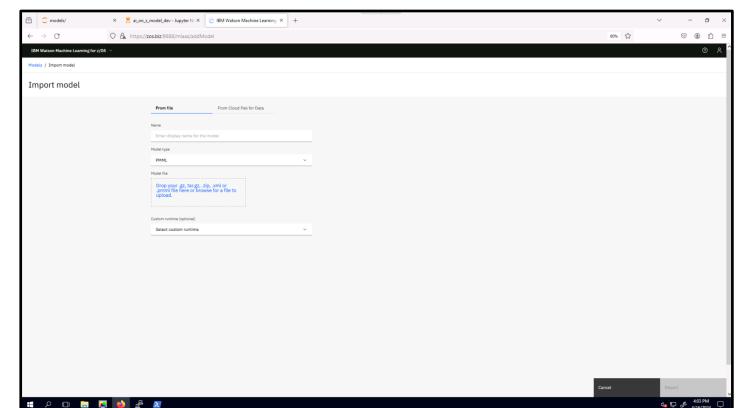
Go to MLz UI

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



1. AI model training

2. AI model deployment

3. Preprocessing pipeline deployment

4. AI model inferencing

[Go to MLZ UI](#)

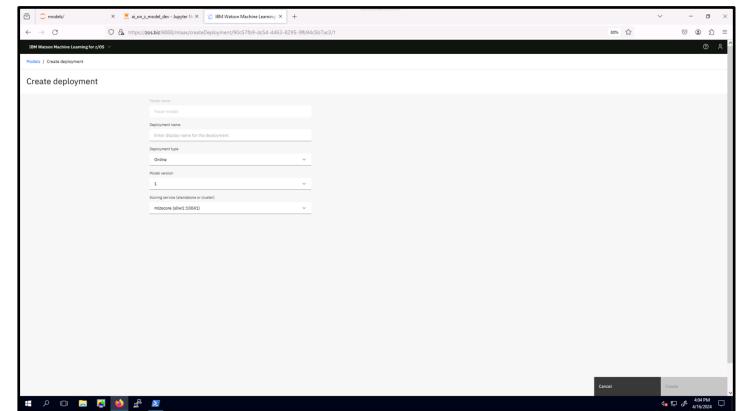
[Import AI model](#)

[Deploy AI model](#)

[View deployed AI model](#)

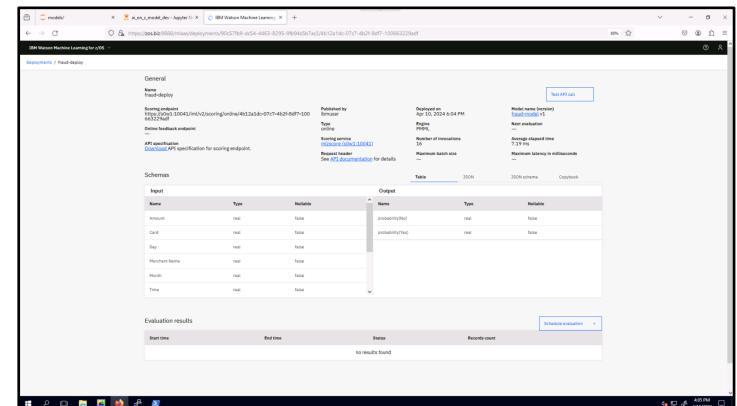
Deploy AI model

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. CICS or REST)
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

3. Preprocessing pipeline deployment

4. AI model inferencing

AI model deployment complete



Prerequisites

- Must have MLz installed

Step 3

Preprocessing deployment

We will deploy our preprocessing pipeline using MLz. We can use this preprocessing pipeline at inference time and send the data to MLz.

1. AI model training

2. AI model deployment

3. Preprocessing pipeline deployment

4. AI model inferencing

[Go to MLz UI](#)

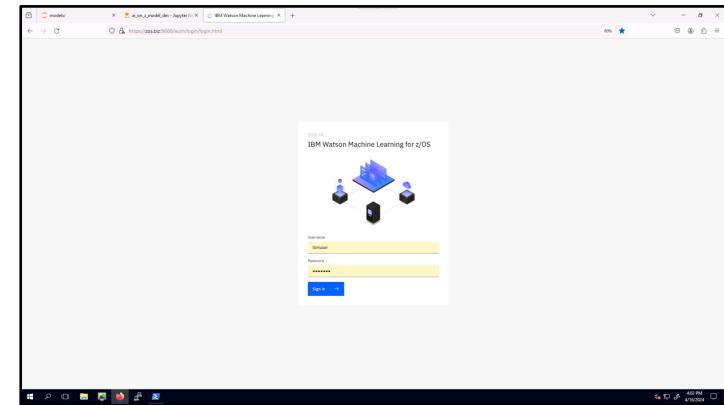
[Import preprocessing pipeline](#)

[Deploy preprocessing pipeline](#)

[View deployed preprocessing pipeline](#)

Go to MLz UI

1. Sign in with username/password



Import preprocessing pipeline

1. Go to models tab
2. Click import model
3. Enter preprocessing pipeline name
4. Choose PMML for model type
5. Drag and drop preprocessing pipeline file
Note: Use your previously trained model
6. Click import

[Go to MLZ UI](#)

[Import preprocessing pipeline](#)

[Deploy preprocessing pipeline](#)

[View deployed preprocessing pipeline](#)

Deploy preprocessing pipeline

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. CICS or REST)
8. Click create

View deployed preprocessing pipeline

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

3. Preprocessing pipeline deployment

4. AI model inferencing

Preprocessing pipeline deployment complete



Prerequisites

- Must have MLz model deployed
- Must have MLz preprocessing pipeline deployed
- Must have Python installed

Step 4

AI model inferencing

We can perform inferencing with our AI model that is deployed within MLz. We will preprocess our data with our deployed MLz preprocessing pipeline and then utilize our deployed AI model for inferencing.

All sample code for this section is within

```
aionz-st-data-preprocessing-mlz/zST-model-inferencing
```

[Configure sample application](#)

[Run inferencing](#)

Configure sample application

- Set environment variables within

```
aionz-st-data-preprocessing-mlz/zST-  
model-inferencing/.env
```

file

- a. MODEL_URL – scoring URL from deployed AI model within MLz
- b. PREPROCESSING_URL – scoring URL from deployed preprocessing pipeline within MLz
- c. MLZ_IP_W_PORT – IP and port of MLz (format is IP:port)
- d. MLZ_USER – MLz username
- e. MLZ_PASS – MLz password

Run inferencing

1. Run sample application

```
python zst_preprocess_inferencing.py
```

1. AI model training

2. AI model deployment

3. Preprocessing pipeline deployment

4. AI model inferencing

AI model inferencing complete