



AI on IBM Z

Health insurance claims 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 health insurance claims solution on IBM Z.

Table of contents

AI model training.....	3
AI model deployment.....	8
AI model integration.....	12



Step 1

AI model training

We will build a health insurance claims 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-health-insurance-claims/zST-model-training-jupyter
```

Prerequisites

- Must have Python (3.9 or 3.10) installed

Dataset guidance

Sample health insurance claims dataset can be found on Kaggle -

https://www.kaggle.com/datasets/grvmishra7/health-insurance-prediction?select=train_Df64byy.csv

Required features

- applicant_id
- years_of_insurance_with_us
- regular_checkup_last_year
- adventure_sports
- occupation
- visited_doctor_last_1_year
- cholesterol_level
- daily_avg_steps
- age
- heart_decs_history
- any_other_major_decs_history gender
- avg_glucose_level
- bmi
- smoking_status
- year_last_admitted
- location
- alcohol
- exercise
- weight_change_in_last_one_year
- fat_percentage
- insurance_cost

1. AI model training.

Access rapid AI on IBM Z development environment

Provide data

Model training

Access trained AI model

2. AI model deployment

Access rapid AI on IBM Z development environment

1. Create and activate Python virtual environment

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

2. Install required Python packages

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

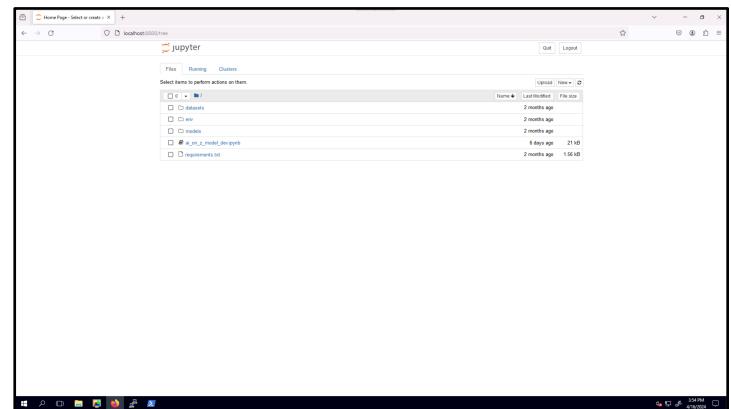
3. Run Jupyter

```
jupyter notebook
```

4. View Jupyter interface

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

4. AI model integration



① 1. AI model training.

Access rapid AI on IBM Z development environment

Provide data

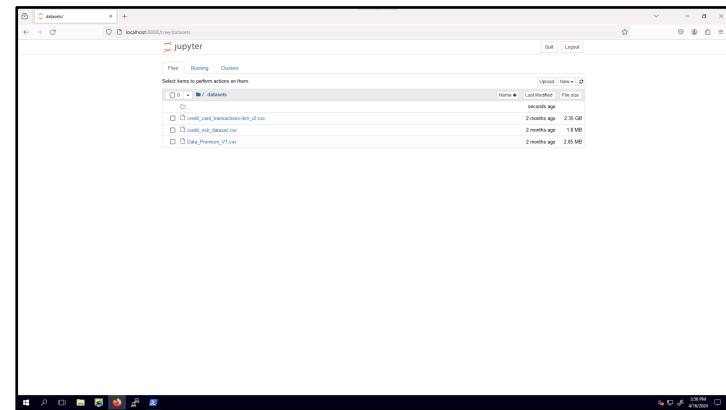
Model training

Access trained AI model

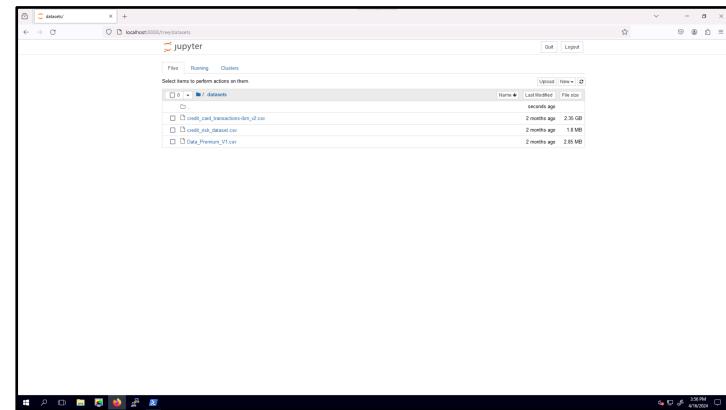
Provide data

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

○ 2. AI model deployment



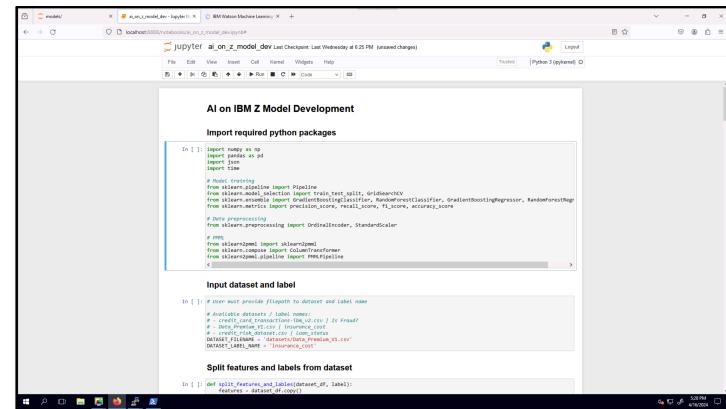
○ 4. AI model integration



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

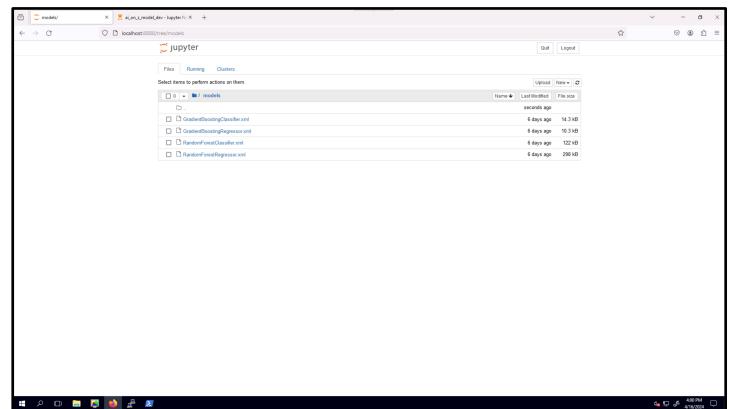
[Access trained AI model](#)

2. AI model deployment

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)

4. AI model integration

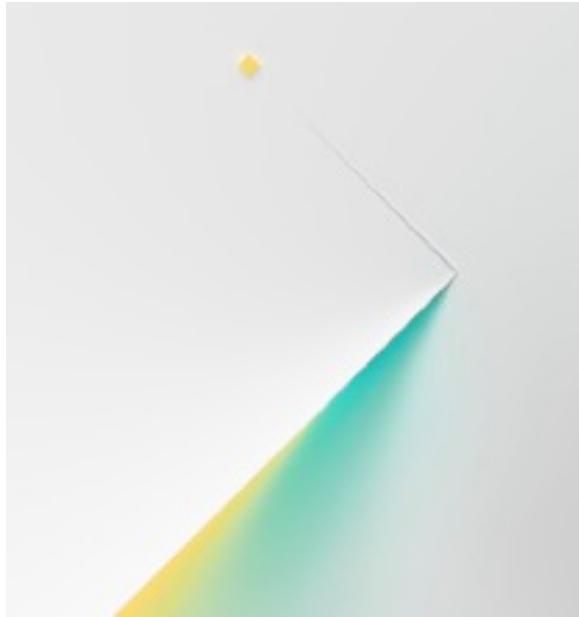


1. AI model training.

2. AI model deployment

4. AI model integration

AI model training complete



Prerequisites

- Must have MLz installed

Step 2

AI model deployment

We will deploy our health insurance 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

4. AI model integration

[Go to MLz UI](#)

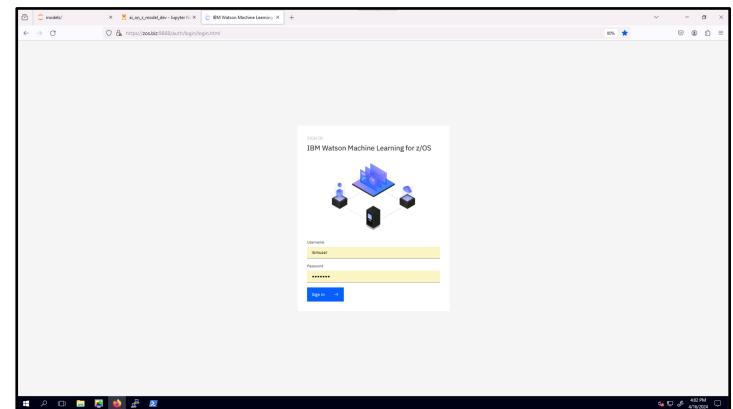
[Import AI model](#)

[Deploy AI model](#)

[View deployed AI model](#)

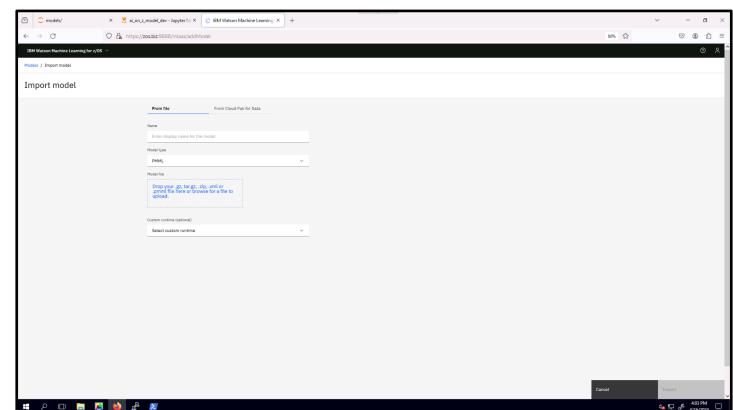
Go to MLz UI

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

4. AI model integration

[Go to MLZ UI](#)

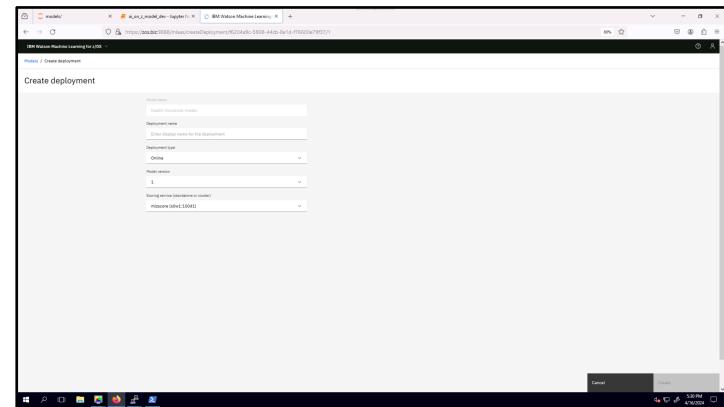
[Import AI model](#)

[Deploy AI model](#)

[View deployed AI model](#)

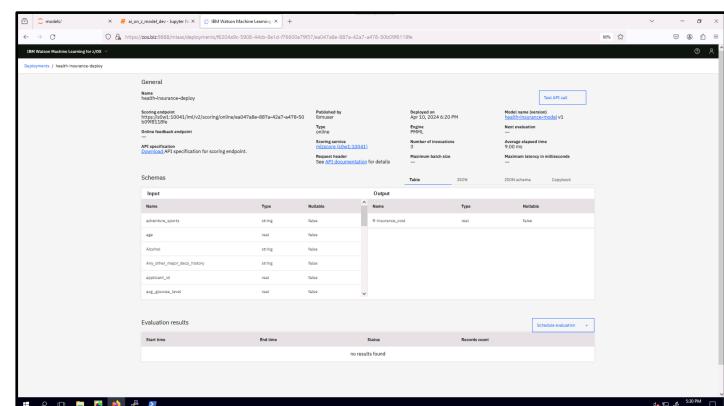
Deploy AI model

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

4. AI model integration

AI model deployment complete



Prerequisites

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

Step 3

AI model integration

We can use our deployed MLz health insurance claims AI model and integrate it into different types of applications. Guidance on integrating the AI model into a sample health insurance claims application is below.

All sample code for this section is within

```
aionz-st-health-insurance-claims/zST-model-integration-HIC
```

1. AI model training.

[Get model details for inferencing](#)

[Configure sample application](#)

Build sample application

Deploy sample application

Access sample application

2. AI model deployment

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

4. AI model integration

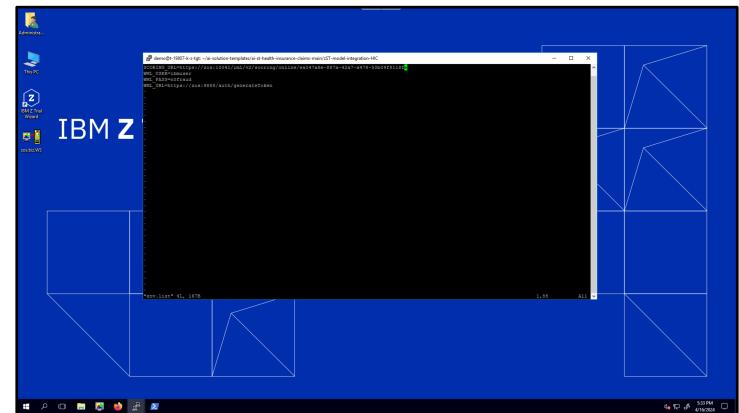
The screenshot shows the 'IBM Watson Machine Learning for z/OS' interface. A modal window titled 'Health-insurance-deploy' is open, displaying model details. The 'General' tab is selected, showing the model name 'Health-insurance-deploy', creation date '2024-01-15T04:16:42Z', and status 'Published'. It also shows the API endpoint 'https://us-south.ml.cloud.ibm.com/ml/v1/deployments/00000000-0000-0000-0000-000000000000'. The 'Schemas' tab shows input and output schema definitions. The 'Evaluation results' section indicates 'no results found'.

Configure sample application

1. Set the environment variables within

```
aionz-st-health-insurance-claims-main/zST-model-integration-HIC/env.list
```

WML_IP_W_PORT (IP address of MLz with port)
WML_USER (MLz username)
WML_PASS (MLz password)
WMLZ_ENDPOINT (scoring endpoint for deployed AI model)



1. AI model training.

2. AI model deployment

4. AI model integration

Get model details for
inferencing

Configure sample
application

[Build sample application](#)

[Deploy sample application](#)

Access sample application

Build sample application

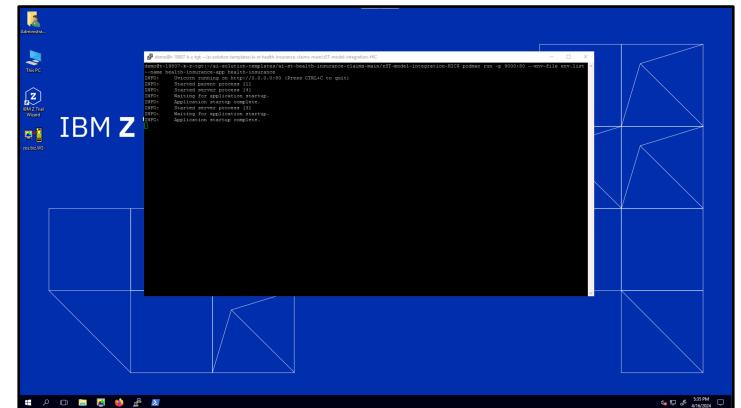
1. Run command in terminal

```
podman build -t health-insurance .
```

Deploy sample application

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

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



1. AI model training.

2. AI model deployment

4. AI model integration

Get model details for inferencing

Configure sample application

Build sample application

Deploy sample application

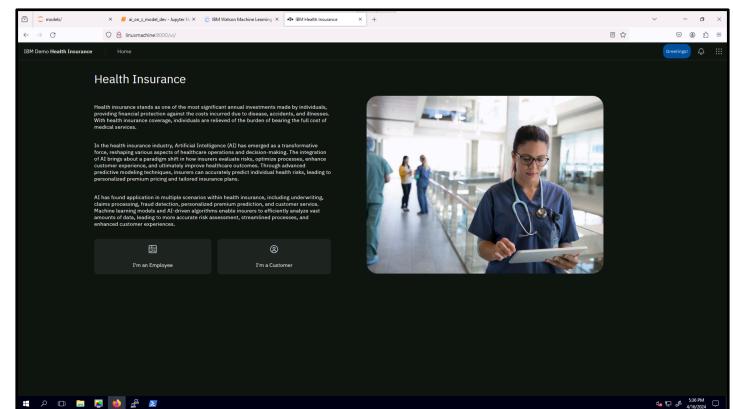
[Access sample application](#)

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



1. AI model training.

2. AI model deployment

4. AI model integration

AI model integration complete