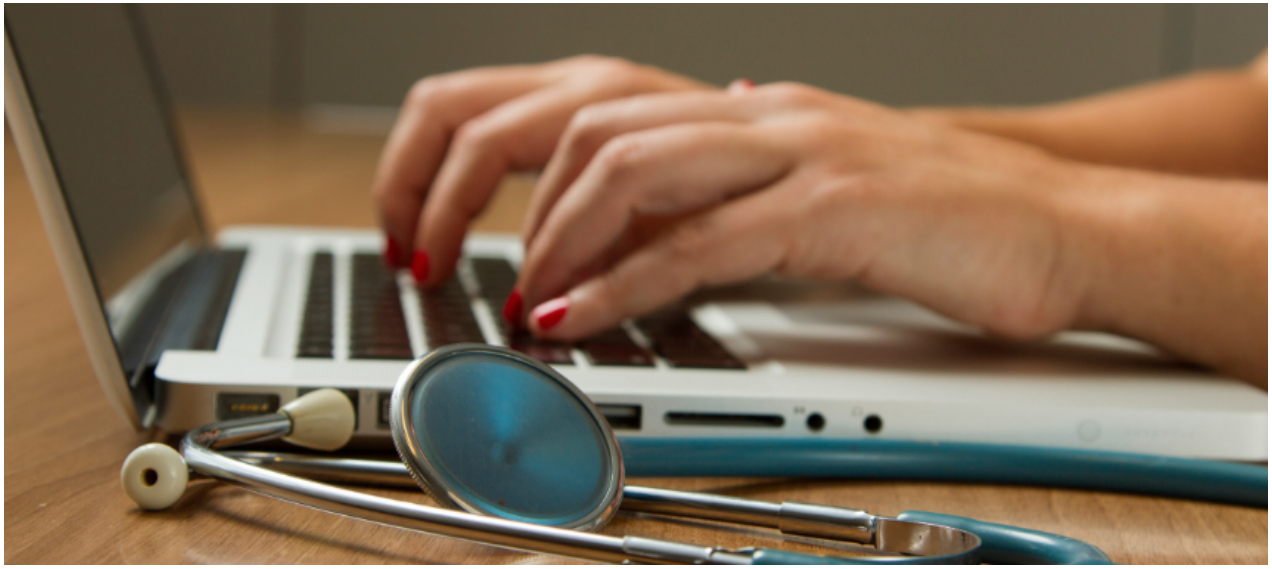


AI on Linux on Z

Health insurance claims solution template

This solution template provides an example on how to deploy AI using a Linux on Z environment, while making use of open source frameworks, Triton Inference Server (TIS), 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 Linux on Z.

Table of contents

Step 1

AI model training

We will build a health insurance claims AI model by training with the provided Rapid AI on Linux on 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-health-insurance-claims-tis/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/gdeepakreddy/insurance?select=Data.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

Access rapid AI on Linux on 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
   ```

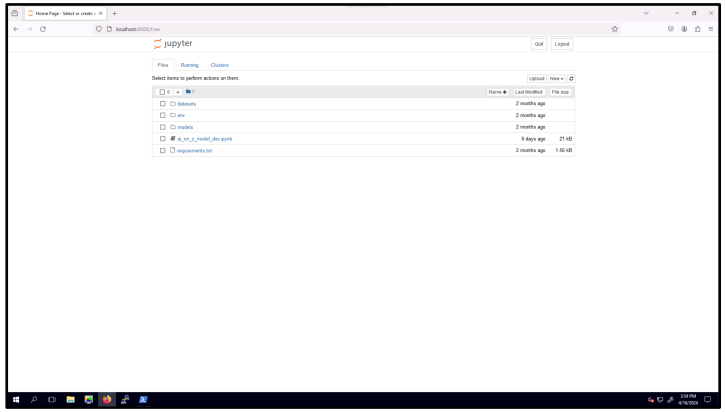
3. Install required Python packages

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

4. Run Jupyter

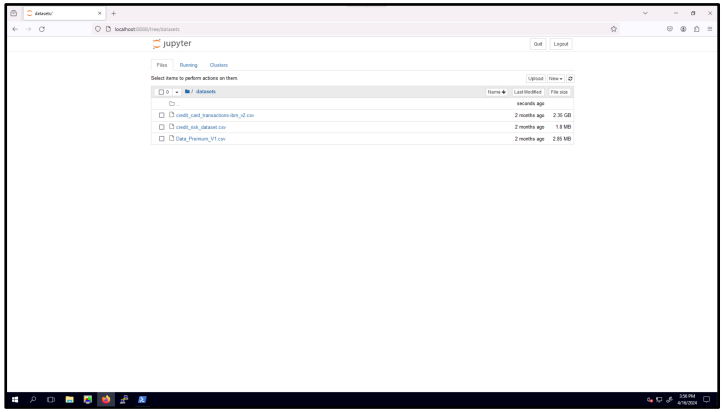
   ```
   jupyter notebook
   ```

5. View Jupyter interface
   a. Go to localhost:8888 in a web browser



Provide data

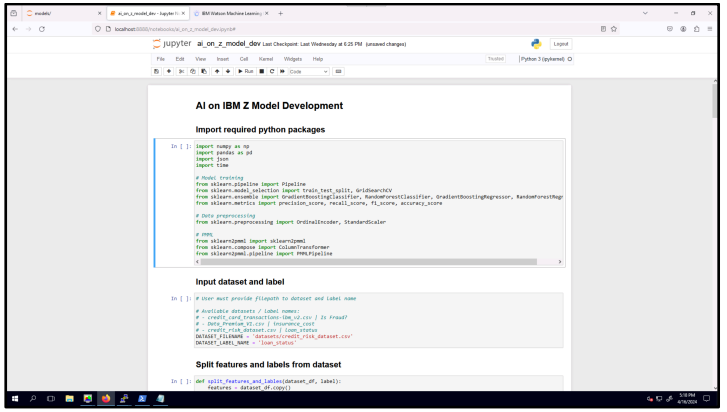
1. Your input dataset (csv) in `datasets/` directory
2. Add input data to Jupyter notebook (ai_on_z_model_dev.ipynb)
    a. Set `DATASET_FILENAME` to the path to your dataset
    b. Set `DATASET_LABEL_NAME` to the name of the column you're predicting from the dataset



Model training

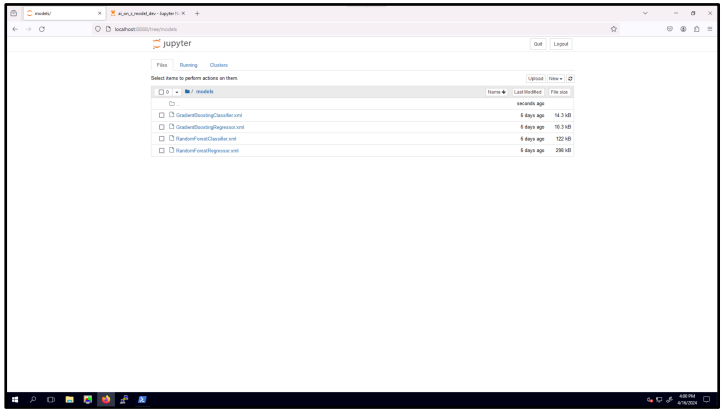
1. Step through and run all cells within Jupyter notebook (ai_on_z_model_dev.ipynb) within web browser

    Note: This may take several minutes



Access trained AI model

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



1. AI model training

2. AI model deployment

3. AI model integration

AI model training complete

Prerequisites

- Must have Docker or Podman installed

Step 2

AI model deployment

We will deploy our health insurance claims AI model using TIS. We can utilize the AI Toolkit to leverage TIS for model deployment. This deployed AI model can then be integrated into applications within the Linux on Z environment.

All sample code for this section is within

`aionz-st-health-insurance-claims-tis/zST-model-deployment`

Build Triton Inference Server

1. Build podman image

   ```
   podman build -t zst-tis .
   ```

Integrate AI model into Triton Inference Server

1. Add your model (.pmml file) to

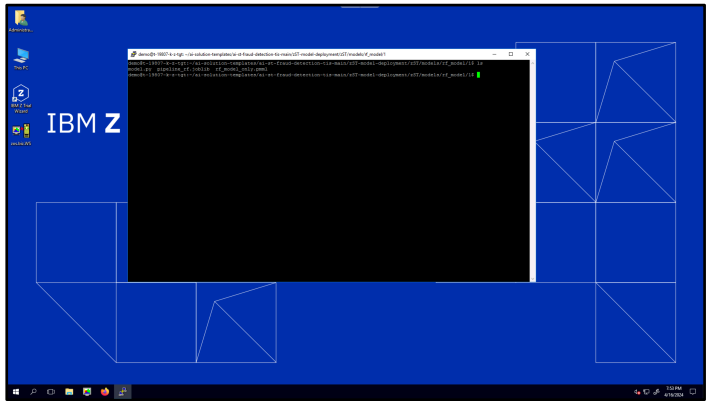
   ```
   aionz-st-health-insurance-claims-tis/zST-model-deployment/zST/models/rf_model/1
   ```

   directory

2. Add your preprocessing .joblib file to

   ```
   aionz-st-health-insurance-claims-tis/zST-model-deployment/zST/models/rf_model/1
   ```

   directory



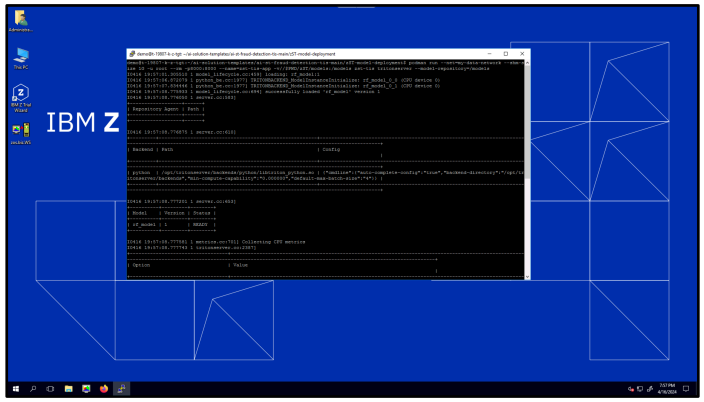
Deploy Triton Inference Server

1. Create docker network

```
podman network create my-data-network
```

2. Run podman container

```
podman run --net=my-data-network --shm-size 1G
-u root --rm -p8000:8000 --name=zst-tis-app -
v//$PWD/zST/models:/models zst-tis
tritonserver --model-repository=/models
```

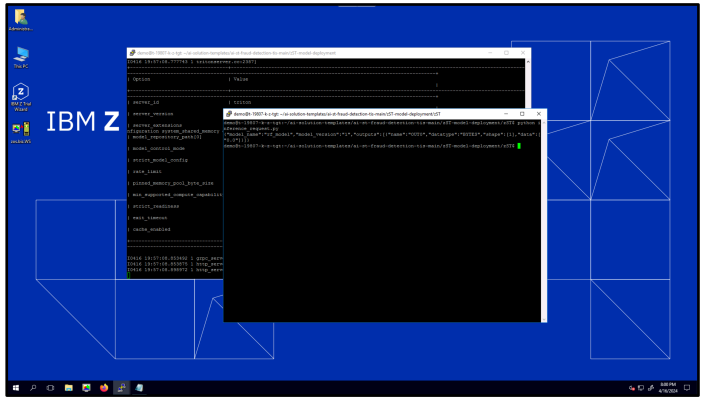


Run sample test

1. Run python script from terminal with ip/port of triton inference server (in new terminal)

```
cd aionz-st-health-insurance-claims-tis/zST-
model-deployment/zST
```

```
python inference_request.py
```



1. AI model training

2. AI model deployment

3. AI model integration

☑ AI model deployment complete

Step 3

AI model integration

We can use our deployed TIS health insurance claims AI model and integrate it into different types of applications. The AI model can be analyzed and/or provide inferencing APIs using the sample AI on Linux on Z health insurance claims dashboard.

All sample code for this section is within

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

Prerequisites

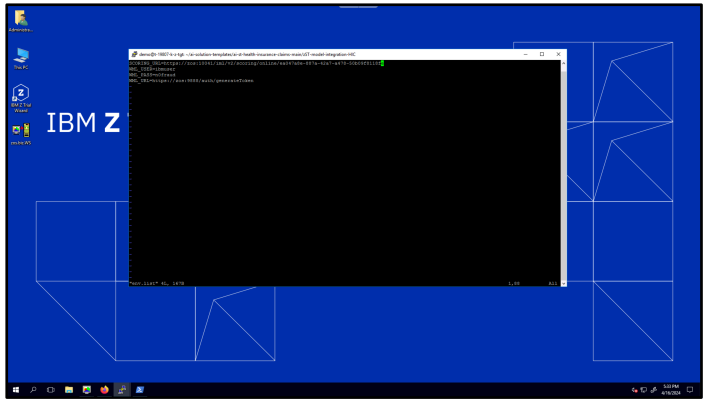
- Must have Docker or Podman installed

Configure sample application

1. Set the environment variables within

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

   TIS_ENDPOINT (scoring endpoint for deployed AI model)



Build sample application

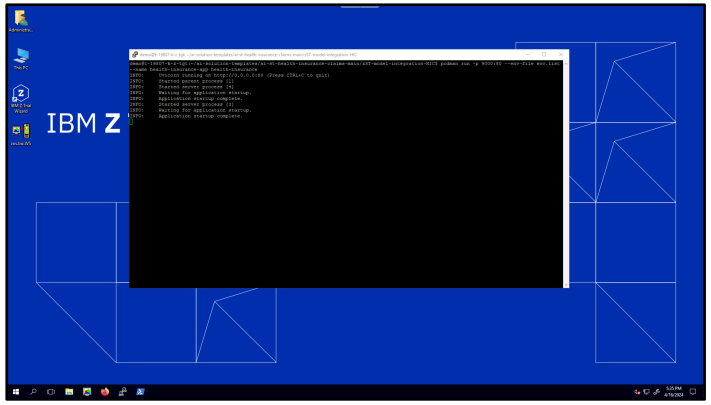
1. Run command in terminal

   ```
   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
   ```

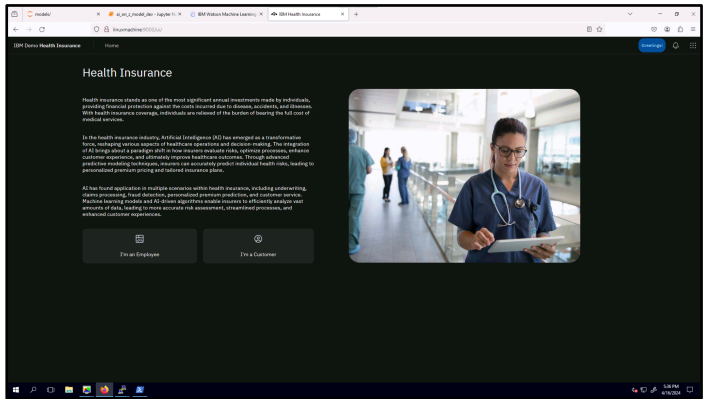


Access sample application

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

3. AI model integration

☑ AI model integration complete