



AI on IBM Z

Credit risk assessment 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, Watson Machine Learning for z/OS (WMLz), 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 credit risk assessment solution on IBM Z.

Table of contents

AI model training.....3

AI model deployment.....7

AI model integration.....11



Step 1

AI model training

We will build a credit risk assessment 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

```
ai-st-credit-risk-assessment/zST-model-training-jupyter
```

Prerequisites

1. Must have Python (3.9 or 3.10) installed

Dataset guidance

Sample open source credit risk assessment dataset can be found on Kaggle -

<https://www.kaggle.com/datasets/laotse/credit-risk-dataset>

Required features

- person_age
- person_income
- person_home_ownership
- person_emp_length
- loan_intent
- loan_grade
- loan_amnt
- loan_int_rate
- loan_status
- loan_percent_income
- cb_person_default_on_file
- cb_person_cred_hist_length

[Access rapid AI on IBM Z development environment](#)

[Provide data](#)

[Model training](#)

[Access trained AI model](#)

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

Go to localhost:8888 in a web browser

5. Click on ai_on_z_model_dev.ipynb in web browser

Provide data

1. Add your input dataset (csv) into datasets/ directory
2. Add input data to Jupyter notebook
 - Set DATASET_FILENAME to the path to your dataset
 - Set DATASET_LABEL_NAME to the name of the column you're predicting from the dataset

```

AI on IBM Z Model Development

Import required python packages

In [ ]: import numpy as np
import pandas as pd
import sys
import time

# Model training
from sklearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import GradientBoostingClassifier, RandomForestClassifier
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score

# Data preprocessing
from sklearn.preprocessing import OrdinalEncoder, StandardScaler

# ERM
from sklearn.pipeline import make_pipeline
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import make_pipeline

Input dataset and label

In [ ]: # User must provide filepath to dataset and label name
DATASET_FILENAME = 'datasets/credit_card_transactions.csv'
DATASET_LABEL_NAME = 'is_fraud?'

Split features and labels from dataset

In [ ]: def split_features_and_labels(dataset_df, label):
    features = dataset_df.copy()
    labels = features.pop(label)
    return features, labels
  
```



[Access rapid AI on IBM Z development environment](#)

[Provide data](#)

[Model training](#)

[Access trained AI model](#)

Model training

1. Step through and run Jupyter notebook from web browser

The screenshot shows a Jupyter Notebook interface with the title 'ai_on_z_model_dev'. The code in the notebook is as follows:

```

import numpy
import pandas
import json
import time

# Model training
from sklearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import GradientBoostingClassifier, RandomForestClassifier
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score

# Data preprocessing
from sklearn.preprocessing import OrdinalEncoder, StandardScaler

# Train
from sklearn.glm import sklearn_glm
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline

# Input dataset and label
# You must provide filepath to dataset and label name
DATASET_FILEPATH = 'dataset/crowdli_xavi_transactions.csv'
DATASET_LABEL_NAME = 'Is Fraud?'

# Split features and labels from dataset
def split_features_and_labels(dataset_filepath, label):
    features = dataset_filepath.copy()
    labels = features.pop(label)
    return features, labels
  
```

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

1. 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.

[Go to MLz UI](#)

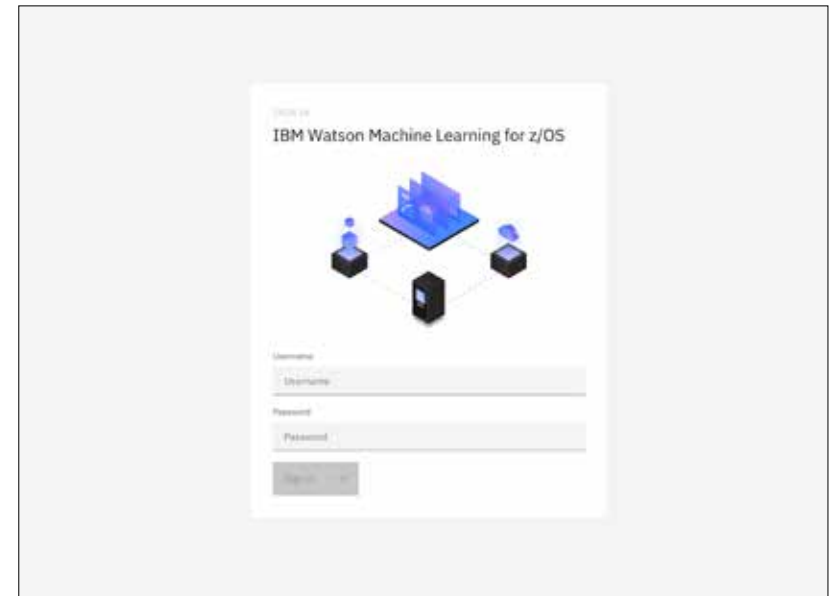
[Import AI model](#)

[Deploy AI model](#)

[View deployed AI model](#)

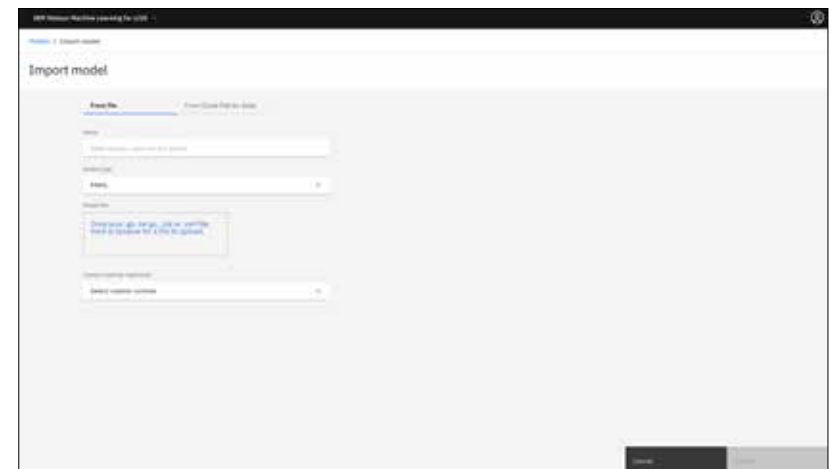
Go to MLz UI

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
Choose PMML if using your previously trained model
5. Drag and drop model file
Use your previously trained model
6. Click import



[Go to MLz UI](#)

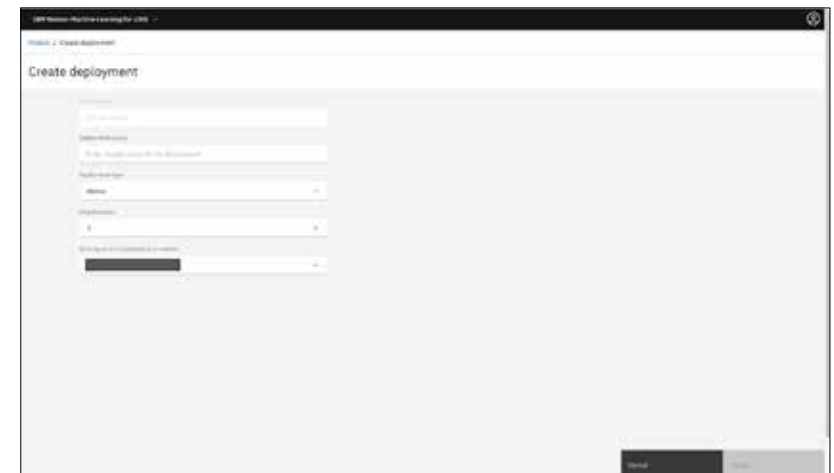
[Import AI model](#)

[Deploy AI model](#)

[View deployed AI model](#)

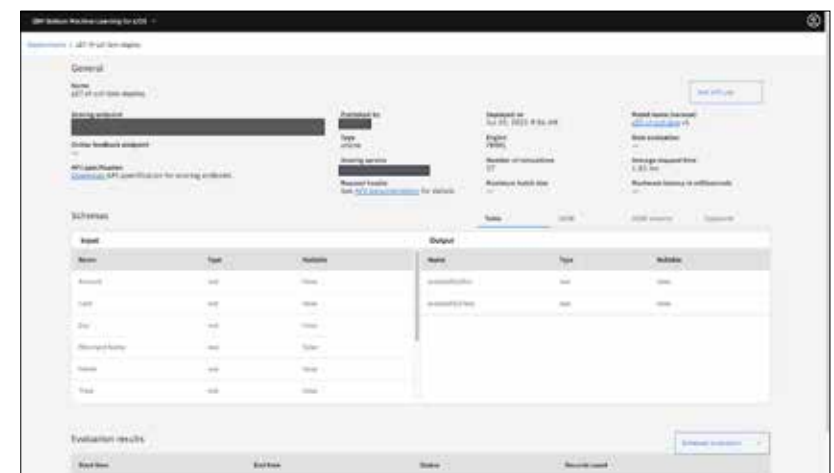
Deploy AI model

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. AI model integration

✓ AI model deployment complete



Prerequisites

1. Must have node.js v16 or newer installed
2. Must have Docker installed
3. Must have Git installed

Step 3

AI model integration

We can use our deployed MLz credit risk assessment AI model and integrate it into different types of applications. Guidance on integrating the AI model into a sample Credit Default Risk Prediction application and CICS-COBOL application are below. The AI model can be analyzed and/or provide inferencing APIs using a the sample AI on IBM Z credit risk assessment Dashboard.

All sample code for this section is within

```
ai-st-credit-risk-assessment/zST-model-integration-cra
```

[Get model details for inferencing](#)

[Configure sample application](#)

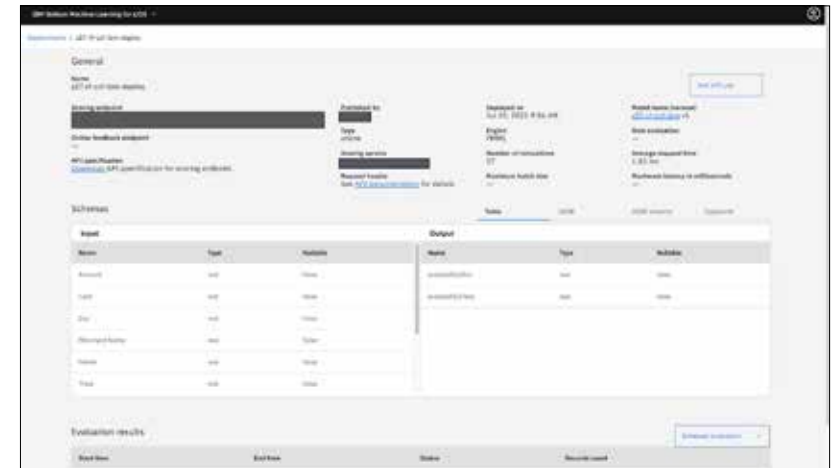
[Build sample application](#)

[Deploy sample application](#)

[Access sample application](#)

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

Set the environment variables within env.list file

- WML_USER (username for MLz)
- WML_PASS (password for MLz user)
- SCORING_URL (scoring endpoint for deployed AI model)

Build sample application

1. Run command in terminal

```
docker build -t credit-risk-assessment .
```

[Get model details for inferencing](#)

[Configure sample application](#)

[Build sample application](#)

[Deploy sample application](#)

[Access sample application](#)

Deploy sample application

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

```
docker run -p 9000:80 --env-file env.list  
--name credit-risk-assessment-app cred-  
it-risk-assessment
```

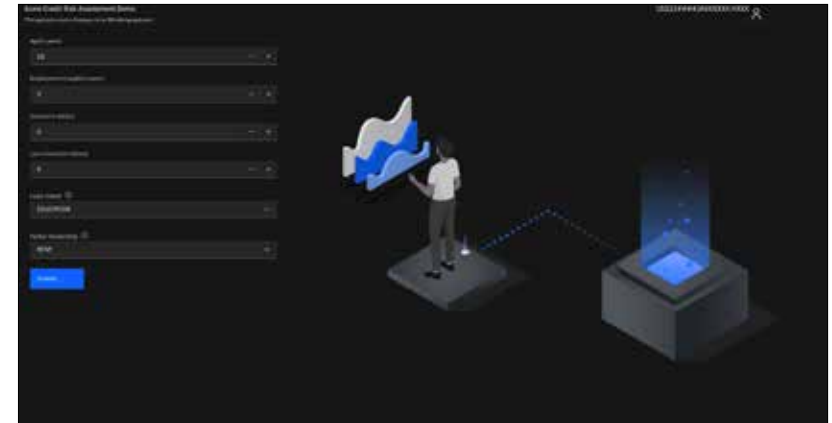
Access sample application

View the following URL in a web browser

- Credit risk assessment
 - <http://{ip address}:{port}///static/cra/>
- Dashboard
 - <http://{ip address}:{port}///static/dashboard/>

Note:

- IP address: IP of server you deployed application in
- Port: port you used with docker run



✓ 1. AI model training

✓ 2. AI model deployment

✓ 3. AI model integration

✓ AI model integration complete