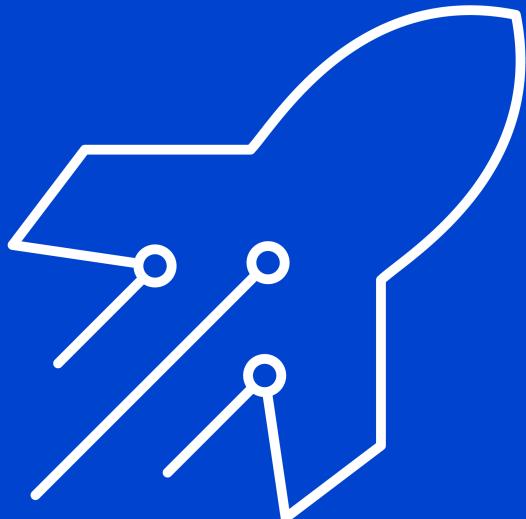


# Lab Guide

## CP4Data and Z Integration HOL

Session: EZS66T316BH

Maggie Lin [meichi@us.ibm.com](mailto:meichi@us.ibm.com), Watson Machine Learning for z/OS Development





## Table of Contents

Legal Disclaimer .....	4
Introduction.....	5
Lab Overview .....	5
Lab Instructions .....	6
CP4D integration with WMLz.....	6
Log in to CP4D.....	6
1. Model Training in CP4D .....	9
Log in to IBM Watson Machine Learning for z/OS.....	19
2. Model Management and Deployment in WMLz .....	22
Section 1. Dashboard.....	22
Section 2. Model Management .....	23
Section 3. Deployment Model to Scoring Server .....	27
We Value Your Feedback!.....	34

## Legal Disclaimer

© IBM Corporation 2020. All Rights Reserved. The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

## Introduction

IBM Watson Machine Learning for z/OS (WMLz) is an enterprise machine learning platform that provides end to end management of the entire machine learning workflow. WMLz helps to simplify and significantly reduce the time to train and deploy machine learning models by:

- Integrating all the tools and functions needed for machine learning and automating the machine learning workflow.
- Providing a platform for better collaboration across different personas including Data Scientist, Data Engineer, Business Analyst and Application Developers, for a successful machine learning project.
- Infusing cognitive capabilities into the machine learning workflow to help determine when model results deteriorate and need to be tuned and provide suggestions for updates or changes.

## Lab Overview

For this hands-on lab, you have the opportunity to use CP4Data and Watson Machine Learning for z/OS (WMLz) to experience the full life cycle of predictive model development and management

- Train a model and save it in CP4D environment
- Import the CP4D trained model to WMLz, and deploy the model using the Model Management Dashboard in WMLz
- Test scoring of the deployed model in WMLz

## [Lab Instructions](#)

Each student should receive a lab worksheet. The worksheet provides information that is specific to each student to use throughout the lab.

Some of the instructions in the lab instruct the student to use a value that is marked with angle brackets (< >). The value the student should use is on their lab worksheet. Make sure to exclude the brackets when using the value for its intended purpose in the lab.

## [CP4D integration with WMLz](#)

This customer churn model is a simplified customer churn model to determine the likelihood of the customer leaves the phone carrier. Every user has their own profile and information like phone usage, payment method, and annual income. The value of adding Machine Learning to the customer churn is that ML generates a smart score for the customer with the information provided. The phone carrier can define a threshold for this score to help determine if there is additional action they can take to retain the customer. The customer churn is a critical metric because it is much less expensive to retain existing customers than it is to acquire new customers.

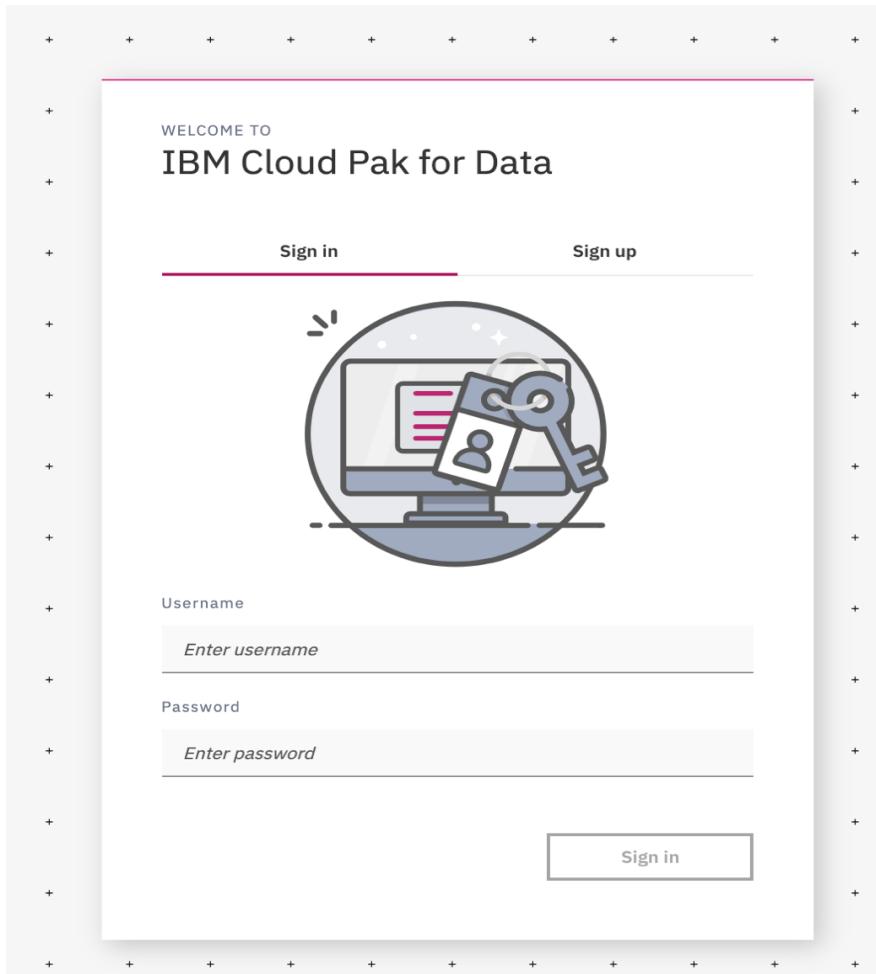
In this session, you will go through the process of building the customer churn model with CP4D, experiencing the flow to analyze data, develop model, and then import the model and deploy model with WMLz.

- Model Training with Jupyter notebook
- Model Management and Deployment

## [Log in to CP4D](#)

The URL and authentication credentials for the CP4D Web UI are provided on a separate worksheet. Contact the lab instructor if you did not receive a worksheet.

- 1. Start the Chrome or Firefox browser, and enter the URL assigned to you, e.g. <cp4d\_login\_url>.
- 2. Enter the <cp4d\_login\_userid> and <cp4d\_login\_password> credentials from your lab worksheet.



You should be presented with the CP4D Lets get started main page

IBM Cloud Pak for Data

All

Search

WELCOME, ctp!

# Let's get started!

Use these resources to make the most of your IBM Cloud Pak for Data experience.

Collect and organize

IBM Cloud Pak for Data:  
Collect and organize

0:00 / 17:44

Build your enterprise data catalog and ensure that your policies and rules.

- Explore business terms
- Explore policies
- Discover assets
- Explore catalogs

DATA JOURNEY

## From data to insight

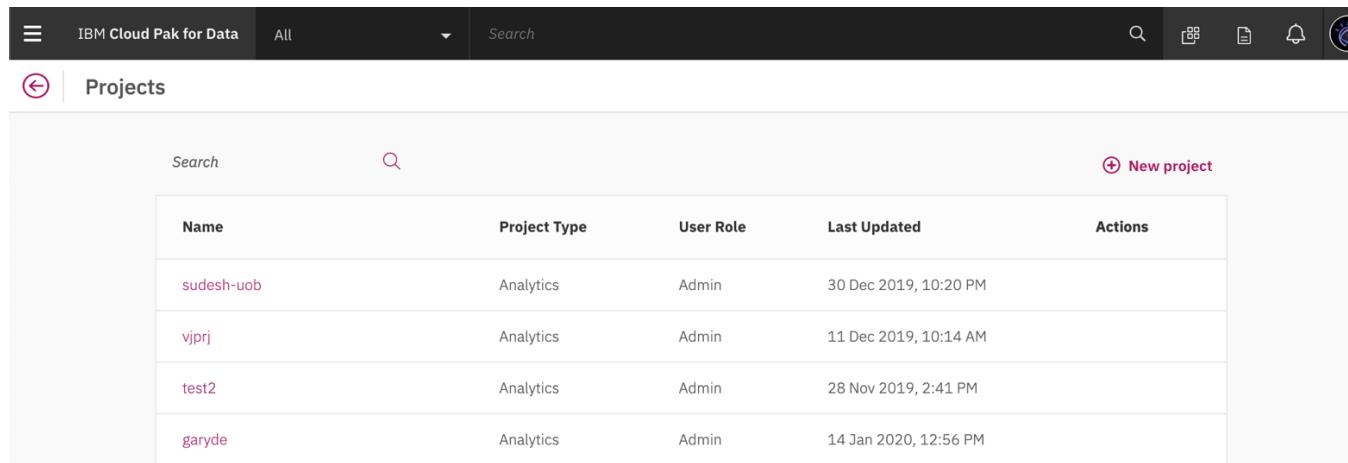
Understand the end-to-end data journey to see how your role fits into the story.

# 1. Model Training in CP4D

In this Lab, we will use the Web UI of CP4D to leverage machine learning to understand customer churn and determine which customer is likely to leave. You will learn how to use Jupyter notebook of CP4D to train a binary classification scikit learn (RandomForestClassifier) model.

- 1. Select a project to contain your notebook.

Click on the menu button on the top-left corner and select “**Projects**” .



The screenshot shows the 'Projects' page in the IBM Cloud Pak for Data interface. At the top, there is a navigation bar with the IBM logo, a search bar, and several icons. Below the navigation bar, the title 'Projects' is displayed next to a back arrow icon. The main area contains a table with the following data:

Name	Project Type	User Role	Last Updated	Actions
sudesh-uob	Analytics	Admin	30 Dec 2019, 10:20 PM	
vjprj	Analytics	Admin	11 Dec 2019, 10:14 AM	
test2	Analytics	Admin	28 Nov 2019, 2:41 PM	
garyde	Analytics	Admin	14 Jan 2020, 12:56 PM	

If you do not have any projects, click “**New Project**” button to create one and name it “<login\_userid>-IMLz\_HOL”. Then click on the project to open the project summary page.

- 2. From the project overview, click on “Asset” and then on the Asset page, click on “New data asset” to add customer.csv and MyLabelEncoder-1.2.0.401.post201912301918.tar.gz

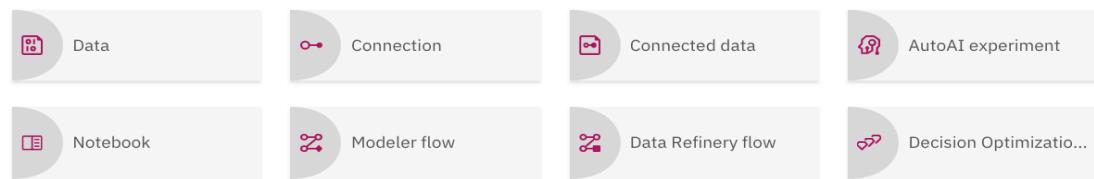
Screenshot of the IBM Cloud Pak for Data interface showing the 'Assets' tab selected. The page displays a list of data assets, including 'customers.csv' and 'MyLabelEncoder-1.2.0.401.post201912301918.tar.gz'.

<input type="checkbox"/>	NAME	TYPE	CREATED BY	LAST MODIFIED
<input type="checkbox"/>	customers.csv	Data Asset	ctp	10 Jan 2020, 12:56:15
<input type="checkbox"/>	MyLabelEncoder-1.2.0.401.post201912301918.tar.gz	Data Asset	ctp	10 Jan 2020, 12:56:01

- 3. Click “Add to project” to add “Notebook” asset to the project.

### Choose asset type

#### AVAILABLE ASSET TYPES



- 4. Create a new notebook “From file”. ChurnScikit\_CP4D\_WMLz.ipynb, then click “Create Notebook” button.

The screenshot shows the 'Add Notebook' page in the IBM Cloud Pak for Data interface. The top navigation bar includes 'IBM Cloud Pak for Data', a search bar, and a 'My Projects' section showing 'forMaggie'. Below this, the page title is 'New notebook' with tabs for 'Blank', 'From file' (which is selected), and 'From URL'. The 'Name' field contains 'ChurnScikit\_CP4D\_WMLz' with 19 characters remaining. The 'Description (optional)' field has a placeholder 'Type your Description here' and 500 characters remaining. Under 'Select runtime', 'Default Python 3.6 (1 vCPU and 2 GB RAM)' is selected. In the 'Notebook file' section, 'ChurnScikit\_CP4D\_WMLz.ipynb' is listed. Below it, there's a note to import a local .ipynb file. At the bottom right are 'Cancel' and 'Create Notebooks' buttons.

New notebook

Blank    From file    From URL

Name

ChurnScikit\_CP4D\_WMLz

19 characters remaining

Description (optional)

Type your Description here

500 characters remaining

Select runtime

Default Python 3.6 (1 vCPU and 2 GB RAM)

Notebook file

ChurnScikit\_CP4D\_WMLz.ipynb

Import a notebook file (.ipynb) from your local device.

Create Notebooks

You will see the following spinner indicating a runtime container is being created for the Jupyter notebook. This could take a minute or two for the first time you open a notebook container in your project.



Starting Runtime for wmlz-churn

The selected runtime has 1 vCPU and 2 GB RAM.

A message indicates the Jupyter notebook is being launched. Shortly afterward, you will see the ChurnScikit\_CP4D\_WMLz.ipynb loaded.

The screenshot shows a Jupyter Notebook interface within the IBM Cloud Pak for Data environment. The top navigation bar includes 'IBM Cloud Pak for Data', 'All', and a search bar. The project path 'My Projects > forMaggie > ChurnScikit\_CP4D\_WMLz' is displayed. The toolbar below includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, and various cell type icons. The main area shows the following code execution:

```
In [1]: # In this Notebook we shall create a Machine Learning Model using Scikit Learn 0.19
There are several different approaches and frameworks for predicting the likelihood of a customer to churn. In this notebook, we purpose.

In [110]: import sklearn
print(sklearn.__version__)
0.19.1

In [3]: !pip install --user scikit-learn==0.19.1
Collecting scikit-learn==0.19.1
  Downloading https://files.pythonhosted.org/packages/3d/2d/9fbc7baa5f44bc9e88ffb7ed32721b879b
earn-0.19.1-cp36-cp36m-manylinux1_x86_64.whl (12.4MB)
| ################################## | 12.4MB 10.7MB/s eta 0:00:01
Installing collected packages: scikit-learn
Successfully installed scikit-learn-0.19.1

In [4]: !pip install /project_data/data_asset/MyLabelEncoder-1.2.0.401.post201912301918.tar.gz
Processing /project_data/data_asset/MyLabelEncoder-1.2.0.401.post201912301918.tar.gz
Building wheels for collected packages: MyLabelEncoder
  Building wheel for MyLabelEncoder (setup.py) ... done
    Created wheel for MyLabelEncoder: filename=MyLabelEncoder-1.2.0.401.post202001102058-cp36-no
3acb3efb7417c46fc8531c544b18e381cb940cb66f0de2d243dce2a6
    Stored in directory: /home/wsuser/.cache/pip/wheels/2b/69/0b/0d1a542a512bea2988b3dcba44df82c
Successfully built MyLabelEncoder
Installing collected packages: MyLabelEncoder
Successfully installed MyLabelEncoder-1.2.0.401.post202001102058

In [111]: import os
import numpy as np
```

Next, we read in a dataset that we will use to develop a Machine Learning model.

We can read the data here in various ways. We are showing here how to read data from a CSV file.

- 5. Now to train the model, execute each notebook cell step by step.  
First, make sure we are using python scikit learn package 0.19.1 version, and install the customized transformer **MyLabelEncoder**.

IBM Cloud Pak for Data All Search

My Projects > forMaggie > ChurnScikit\_CP4D\_WMLz

File Edit View Insert Cell Kernel Widgets Help

Run

In [1]: *# In this Notebook we shall create a Machine Learning Model using Scikit Learn 0.19*

There are several different approaches and frameworks for predicting the likelihood of a customer to churn. In this notebook, purpose.

In [110]: `import sklearn  
print(sklearn.__version__)`  
0.19.1

In [3]: `!pip install --user scikit-learn==0.19.1`

```
Collecting scikit-learn==0.19.1
  Downloading https://files.pythonhosted.org/packages/3d/2d/9fbc7baa5f44bc9e88ffb7ed32721b879
  earn-0.19.1-cp36-cp36m-manylinux1_x86_64.whl (12.4MB)
    |#####| 12.4MB 10.7MB/s eta 0:00:01
  Installing collected packages: scikit-learn
  Successfully installed scikit-learn-0.19.1
```

In [4]: `!pip install /project_data/data_asset/MyLabelEncoder-1.2.0.401.post201912301918.tar.gz`

```
Processing /project_data/data_asset/MyLabelEncoder-1.2.0.401.post201912301918.tar.gz
Building wheels for collected packages: MyLabelEncoder
  Building wheel for MyLabelEncoder (setup.py) ... done
    Created wheel for MyLabelEncoder: filename=MyLabelEncoder-1.2.0.401.post202001102058-cp36-n
  3acb3e1fb7417c46fc8531c544b18e381cb940cb66f0de2d243dce2a6
    Stored in directory: /home/wsuser/.cache/pip/wheels/2b/69/0b/0d1a542a512bea2988b3dcba44df82
  Successfully built MyLabelEncoder
  Installing collected packages: MyLabelEncoder
  Successfully installed MyLabelEncoder-1.2.0.401.post202001102058
```

— 6. Load the training data **customer.csv** and split training data.

The screenshot shows a Jupyter Notebook interface within the IBM Cloud Pak for Data environment. The top navigation bar includes 'IBM Cloud Pak for Data', 'All', and a search bar. Below the toolbar, the path 'My Projects > forMaggie > ChurnScikit\_CP4D\_WMLz' is visible. The toolbar contains standard Jupyter Notebook icons for file operations, cell selection, and help.

In [111]:

```
import os
import numpy as np
```

Next, we read in a dataset that we will use to develop a Machine Learning model.

We can read the data here in various ways. We are showing here how to read data from a CSV file.

In [112]:

```
import pandas as pd
df_data_1 = pd.read_csv('/project_data/data_asset/customers.csv')
df_data_1.head()
```

Out[112]:

ID	LONGDISTANCE	INTERNATIONAL	LOCAL	DROPPED	PAYMETHOD	LOCALBILLTYPE	LONGDISTANCEBILLTYPE
0	1	23	0	206	0	CC	Budget
1	6	29	0	45	0	CH	FreeLocal
2	8	24	0	22	0	CC	FreeLocal
3	11	26	0	32	1	CC	Budget
4	17	12	0	46	4	CC	FreeLocal

In [113]:

```
cmergedDf = df_data_1.copy()
```

In [114]:

```
# drop the ID field
cmergedDf.drop(['ID'], axis=1, inplace=True)
```

In [115]:

```
cmergedDf.head()
```

Out[115]:

	LONGDISTANCE	INTERNATIONAL	LOCAL	DROPPED	PAYMETHOD	LOCALBILLTYPE	LONGDISTANCEBILLTYPE
0	23	0	206	0	CC	Budget	Intl_discount
1	29	0	45	0	CH	FreeLocal	Standard
2	24	0	22	0	CC	FreeLocal	Standard
3	26	0	32	1	CC	Budget	Standard
4	12	0	46	4	CC	FreeLocal	Standard

The screenshot shows a Jupyter Notebook interface within the IBM Cloud Pak for Data environment. The top navigation bar includes 'IBM Cloud Pak for Data', 'All', and a search bar. Below the menu bar, the current project path is 'My Projects > forMaggie > ChurnScikit\_CP4D\_WMLz'. The toolbar contains standard options like File, Edit, View, Insert, Cell, Kernel, Widgets, Help, and various cell execution and format controls.

```

In [116]: cmergedDf.columns.tolist()
Out[116]: ['LONGDISTANCE',
 'INTERNATIONAL',
 'LOCAL',
 'DROPPED',
 'PAYMETHOD',
 'LOCALBILLTYPE',
 'LONGDISTANCEBILLTYPE',
 'USAGE',
 'RATEPLAN',
 'CHURN',
 'GENDER',
 'STATUS',
 'CHILDREN',
 'ESTINCOME',
 'CAROWNER',
 'AGE']

In [117]: from sklearn.model_selection import train_test_split
y = cmergedDf['CHURN']
x = cmergedDf.drop('CHURN', axis=1)
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)

In [118]: from sklearn.pipeline import Pipeline
#from sklearn.preprocessing import StandardScaler, OneHotEncoder, LabelEncoder
from sklearn.preprocessing import LabelEncoder

#categorical_transformer = Pipeline(steps=[('onehot', OneHotEncoder(handle_unknown='ignore'))])
categorical_features = cmergedDf.select_dtypes(include=['object']).drop(['CHURN'], axis=1)

In [119]: cat_indices = [cmergedDf.columns.get_loc(c) for c in categorical_features]

In [120]: df1 = cmergedDf.copy()

In [121]: df1.head()
Out[121]: LONGDISTANCE INTERNATIONAL LOCAL DROPPED PAYMETHOD LOCALBILLTYPE LONGDISTANCEBILLTYPE

```

7. Use RandomForestClassifier as the algorithm for the pipeline training.

IBM Cloud Pak for Data    All    Search

My Projects > forMaggie > ChurnScikit\_CP4D\_WMLZ

File Edit View Insert Cell Kernel Widgets Help

Run Format Code nbdiff

```
In [124]: from sklearn.ensemble import RandomForestClassifier
rf = Pipeline(steps=[ ('cat',categorical_transformer),
                      ('classifier', RandomForestClassifier())])

In [125]: print(rf._final_estimator)

RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=None, max_features='auto', max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
                       oob_score=False, random_state=None, verbose=0,
                       warm_start=False)

In [126]: rf.fit(X_train, y_train)

Out[126]: Pipeline(memory=None,
                    steps=[('cat', Pipeline(memory=None,
                                           steps=[('encoder', <MyLabelEncoder.MyLabelEncoder object at 0x7ff6ba428f28>)]),
                                         trap=True, class_weight=None, criterion='gini',
                                         max_depth=None, max_features='auto', max_leaf_nodes=None,
                                         min_n_estimators=1,
                                         oob_score=False, random_state=None, verbose=0,
                                         warm_start=False))]

In [127]: y_pred = rf.predict(X_test)

In [128]: print(y_pred)

['F' 'F' 'F' 'T' 'F' 'T' 'T' 'T'
 'F' 'F' 'T' 'F' 'T'
 'T' 'T' 'F' 'T' 'F' 'F' 'F' 'T' 'F' 'F' 'T' 'F' 'F' 'T' 'F' 'T' 'F' 'T' 'F' 'F'
 'F' 'F' 'F' 'F' 'T' 'F' 'F' 'F' 'F' 'T' 'F' 'F' 'T' 'F' 'T' 'F' 'F' 'T' 'F' 'F'
 'T' 'T' 'T' 'F' 'T' 'F' 'F' 'T' 'T' 'F' 'F' 'F' 'F' 'T' 'F' 'F' 'T' 'T' 'T' 'F'
 'T' 'F' 'T' 'T' 'F' 'F' 'T' 'T' 'F' 'F' 'T' 'F' 'F' 'T' 'F' 'F' 'T' 'F' 'F' 'T'
 'T' 'F' 'F' 'T' 'F' 'F' 'F' 'T' 'F' 'F' 'T' 'F' 'F' 'T' 'F' 'F' 'T' 'F' 'F' 'T'
 'F' 'F' 'F' 'F' 'T' 'F' 'F' 'F' 'F' 'T' 'F' 'F' 'T' 'F' 'F' 'F' 'F' 'F' 'T' 'F'
 'T' 'T' 'F' 'F' 'F' 'T' 'F' 'T' 'F' 'T' 'T' 'T' 'F' 'F' 'F' 'F' 'T' 'F' 'T' 'F'
 'T' 'F' 'T' 'F' 'F' 'F' 'F' 'F' 'F' 'T' 'F' 'F' 'T' 'F' 'F' 'F' 'F' 'F' 'F' 'T'
```

- 8. Last part to import **WatsonMachineLearningAPIClient** package, use default space id to save the model.

IBM Cloud Pak for Data All Search

My Projects > forMaggie > ChurnScikit\_CP4D\_WMLz

File Edit View Insert Cell Kernel Widgets Help

Run Format Markdown nbdiff

```
In [ ]: from watson_machine_learning_client import WatsonMachineLearningAPIClient
In [ ]: token = os.environ['USER_ACCESS_TOKEN']
wml_credentials = {
    "token": token,
    "instance_id": "wml_local",
    "url": os.environ['RUNTIME_ENV_APXS_URL'],
    "version": "2.5.0"
}
In [ ]: client = WatsonMachineLearningAPIClient(wml_credentials)
In [ ]: space_id = client.spaces.store(meta_props={client.spaces.ConfigurationMetaNames.NAME: "ChurnScikitTrained@CP4D", client.repository.ModelMetaNames.TYPE: "scikit-learn_0.19", client.repository.ModelMetaNames.RUNTIME_UID: "scikit-learn_0.19-py3.6", client.repository.ModelMetaNames.SPACE_UID: space_id})
In [ ]: client.set.default_space(space_id)
In [ ]: print(space_id)
In [ ]: metadata = {
    client.repository.ModelMetaNames.NAME: "churn-scikit-trained@CP4D",
    client.repository.ModelMetaNames.TYPE: "scikit-learn_0.19",
    client.repository.ModelMetaNames.RUNTIME_UID: "scikit-learn_0.19-py3.6",
    client.repository.ModelMetaNames.SPACE_UID: space_id
}
In [ ]: model_artifact = client.repository.store_model(rf, meta_props=metadata, training_data=x_train, scoring_parameters=y_train)
In [ ]:
```

You complete the first part of the lab to get the model trained and saved successfully in CP4d.

## [Log in to IBM Watson Machine Learning for z/OS](#)

The URL and authentication credentials for the WMLz Web UI are provided on a separate worksheet. Contact the lab instructor if you did not receive a worksheet.

- 9. Start the Chrome or Firefox browser, and enter the URL assigned to you, e.g. <login\_url>. For the first-time logon, you are likely to be prompted with a message regarding the security of the page you are visiting. Click “Advance” and accept the exception and proceed with the intended URL.

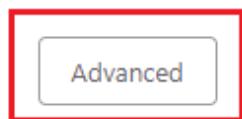


### Your connection is not private

Attackers might be trying to steal your information from **mlzdl03** (for example, passwords, messages, or credit cards). [Learn more](#)

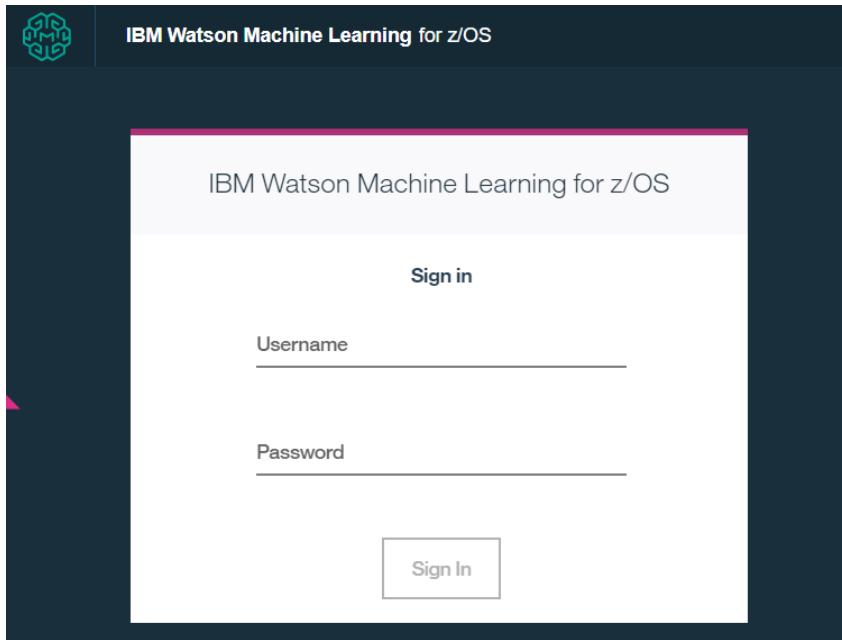
NET::ERR\_CERT\_AUTHORITY\_INVALID

Help improve Safe Browsing by sending some [system information and page content](#) to Google.  
[Privacy policy](#)



[Back to safety](#)

- 10. Enter the <login\_userid> and <login\_password> credentials from your lab worksheet.



You should be presented with the WMLz main page

A screenshot of the WMLz main page. The top navigation bar includes the IBM Watson logo, the text "IBM Watson Machine Learning for z/OS", and a dropdown menu. The main content area is titled "Getting started" and features four buttons: "New project" (document icon), "New notebook" (notebook icon), "New data visualization" (chart icon), and "New modeler flow" (flowchart icon). Below this is a section titled "Recent projects" with a "View all" link and a "My P" link. A table lists one recent project: "mlz-samples" (Project Type: -, Role: Viewer, Collaborators: -, LAS: 201). At the bottom is a section titled "Sample notebooks" with a "View all" link, showing four thumbnail images of notebooks.

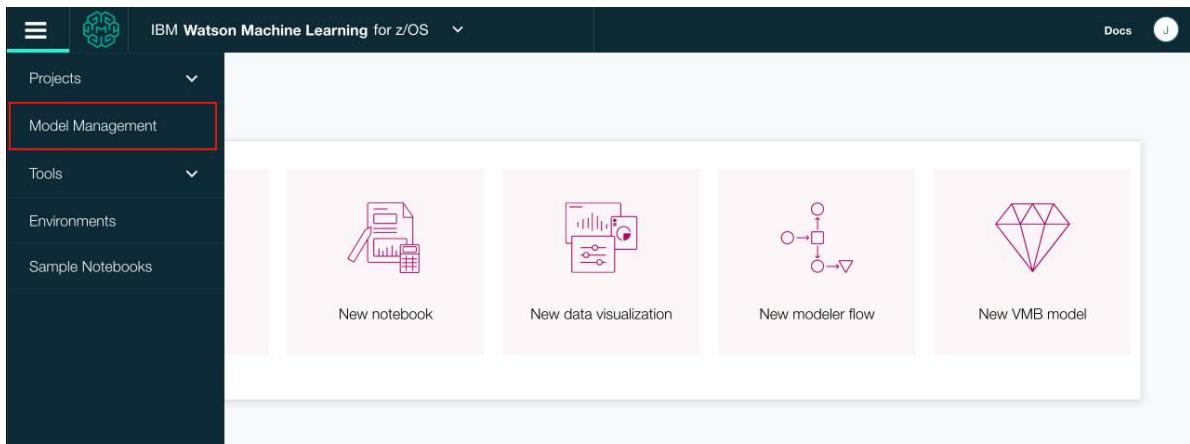


## 2. Model Management and Deployment in WMLz

In this part of the Lab, you will learn as a z/OS System Administrator, how you can use WMLz to view the details of the model that your Data Scientist has created, manage the deployment of the model on your production environment, schedule the model evaluation and test the model using online RESTful API and work with your Application Developer to use the model for scoring and prediction.

### Section 1. Dashboard

- 1. Open the WMLz Dashboard view UI by clicking hamburger menu in the top left and selecting “**Model Management**” as below.



- 2. The “**Dashboard**” shows the statistical information of IBM Machine Learning platform, including the model performance and overall status.

**Current Model Metrics**

0% Performance

- 1 Models accurate
- 4 Models warnings
- 12 Models errors
- 6656 Models unevaluated

**Model with Warnings** See All Models (6673) 1 week

MODEL	EVALUATED VERSION	PUBLISHER	LAST EVALUATION	WARNINGS
IML_NB_Sanity_2019-01-31...	v1	wmlz11	Jan 31, 2019 11:59 AM	1
IML_VMB_XJ_Regression	v1	wmlz11	Feb 4, 2019 5:37 PM	2
IML_VMB_Sanity_2019-01-20...	v1	wmlz11	Feb 4, 2019 3:20 PM	1
IML_VMB_Sanity_2019-01-23...	v1	wmlz11	Feb 5, 2019 8:22 AM	2
IML_NB_Sanity_2019-01-20...	v2	wmlz11	Feb 4, 2019 3:25 PM	2

**Top Deployments by API Calls** 1 week

DEPLOYMENT	DATE DEPLOYED	MODEL	DEPLOYER	API CALLS
jason_test_test	Jan 31, 2019 6:07 AM	jason_test	wmlz11	3
NB-XGBClassifierPipelineG...	Jan 30, 2019 8:41 AM	IML_NB_Sanity_2019-01-30...	wmlz11	2

## Section 2. Model Management

- 3. Click the “**Models**” tab. The “**Models**” page is where you list and manage the Machine Learning models that have been created and exist in your environment. You can view the model details and model versions, retrain, delete and create deployment for a certain model. It also provides a filter to help you locate a specific model.

**Published Models**

Dashboard Models Deployments Data Sources

Import model

MODEL NAME	PUBLISHER	DATE PUBLISHED	MODEL TYPE	LATEST VERSION	EVALUATOR	ACTIONS
jerry-smartloan	jerry	Feb 5, 2019 11:45 AM	PMML	v1	—	⋮

1 models in total 1 of 1 pages First Pre 1 Next Last

- 4. Click on “**Import model**” to import models that are trained outside of WMLz. You see import From File or **From Cloud Pak 4 Data** option.  
Choose **From Cloud Pak 4 Data** option, fill in the CP4D host, username, password.

The screenshot shows the 'Import model' interface for IBM Watson Machine Learning for z/OS. The top navigation bar includes a menu icon, the title 'IBM Watson Machine Learning for z/OS', and a dropdown arrow. Below the title, the breadcrumb path 'Published Models > Import Model' is visible. The main section is titled 'Import model'. There are two tabs at the top: 'From File' and 'From Cloud Pak for Data', with 'From Cloud Pak for Data' being the active tab. A yellow box highlights the 'CP4D Host' input field, which contains the value 'mlpattern.184.170.232.151.nip.io'. To the right of this field is a character count indicator '48'. Below this are fields for 'User Name' (containing 'ctp') with a character count '47' and 'Password' (containing '\*\*\*\*\*') with a character count '44'. At the bottom, a 'Deployment space' dropdown menu is open, showing the placeholder 'Select deployment space'. On the far right, there are 'Cancel' and 'Import' buttons.

- 5. Once it is connected to the CP4D instance, the deployment space drop-down has a list of spaces that is available for you to choose.  
Choose **ChurnScikit\_CP4D\_2\_WMLz** from the list.

IBM Watson Machine Learning for z/OS

Published Models > Import Model

## Import model

From File      **From Cloud Pak for Data**

**CP4D Host \***  
mlpattern.184.170.232.151.nip.io  
48

**User Name \***  
ctp  
47

**Password \***  
\*\*\*\*\*  
44

**Deployment space \***  
Select deployment space ^

PruebaSalesECDeplmtSpace  
ChurnScikit\_CP4D\_2\_WMLz  
checkpyv2  
NA Test

Cancel

From File      **From Cloud Pak for Data**

**CP4D Host \***  
mlpattern.184.170.232.151.nip.io  
48

**User Name \***  
ctp  
47

**Password \***  
\*\*\*\*\*  
44

**Deployment space \***  
Select deployment space ^

PruebaSalesECDeplmtSpace  
ChurnScikit\_CP4D\_2\_WMLz  
checkpyv2  
NA Test

Cancel

- 6. Once ChurnScikit\_CP4D\_2\_WMLz deployment space is selected, go to the **Model name** drop-down list. Choose the mode **churn-scikit-trained@CP4D** model to be imported.

IBM Watson Machine Learning for z/OS

Published Models > Import Model

## Import model

From File      **From Cloud Pak for Data**

**CP4D Host \***  
mlpattern.184.170.232.151.nip.io  
48

**User Name \***  
ctp  
47

**Password \***  
\*\*\*\*\*  
44

**Deployment space \***  
ChurnScikit\_CP4D\_2\_WMLz  
▼

**Model name \***  
Select model name  
churn-scikit-trained@CP4D  
Select model name  
Cancel

- 7. Once you import the model, it takes you back to the Mode list page where you can see your imported model appear in the list.

MODEL NAME	PUBLISHER	DATE PUBLISHED	MODEL TYPE	LATEST VERSION	EVALUATE
June22	jamespак	Jan 14, 2020 11:47 AM	scikit-learn	v1	—
IMPORTED_June21_jamespак_1579031205891	jamespак	Jan 14, 2020 11:46 AM	scikit-learn	v1	—
IMPORTED_June21_jamespак_1579031051134	jamespак	Jan 14, 2020 11:44 AM	scikit-learn	v1	—
churn-scikit-trained@CP4D	mlin	Jan 10, 2020 10:26 PM	scikit-learn	v3	—
CarsModelScala-ByMaggie	mlin	Jan 7, 2020 11:49 AM	SparkML	v3	—
v210_scikit_test	jamespак	Jan 2, 2020 2:41 PM	scikit-learn	v1	—
June21	mlztest	Dec 30, 2019 7:38 PM	scikit-learn	v1	—
churnQin-z-4	mlztest	Dec 20, 2019 11:02 AM	IBM SparkML	v1	—
June21_v210	mlztest	Dec 7, 2019 11:04 AM	scikit-learn	v1	—
importFromJames1202	mlin	Nov 26, 2019 2:40 PM	SparkML	v1	—

### Section 3. Deployment Model to Scoring Server

As a System Administrator, you will need to deploy the model to your z/OS environment and test online. It may sound like a daunting task, but it can be accomplished easily in the WMLz framework:

#### 8. Deploy the model

- Click on the models tab to show the models page. You should already be on this model list page.
- On the row where the model **churn-scikit-trained@CP4D** is, click the actions menu and select “Deploy” button under “ACTIONS.”

You also have the option to deploy a model created by others.

MODEL NAME	PUBLISHER	DATE PUBLISHED	MODEL TYPE	LATEST VERSION	EVALUATOR	ACTIONS
jerry-smartloan	jerry	Feb 5, 2019 11:45 AM	PMML	v1	—	⋮

1 models in total

1 of 1 pages First View details Deploy Setup deployment Retrain Delete Publish to WML Export latest version

- 9. On the “**Create deployment**” page, enter the name of the deployment and select “**Online**” for the **Type**. Choose the scoring service whose name ends in “scoring-cp4d”, and then click the “**Create**” button to save.

Model Name  
jerry-smartloan

Deployment name  
jerry-smartloan-dep

Deployment type  
Online

Engine  
PMML

Model Version  
1

Scoring Service  
mlz16 (9.30.128.39:30163)

Cancel Create

Wait until the deployment completes successfully. Then locate your deployment under the “**Deployments**” page. The “**Deployments**” page is where you view the list of models that have been deployed to your z/OS environment for scoring and prediction. You can test the model here, manage the deployment and schedule evaluations to check the performance of the model on a regular basis.

The screenshot shows the 'Deployments' page of the IBM Watson Machine Learning for z/OS interface. The search bar at the top contains the text 'jerry'. The table below lists one deployment:

DEPLOYMENT NAME	DEPLOYER	MODEL NAME	SCORING SERVICE	TYPE	ENGINE	DATE DEPLOYED	NEXT EVALUATION	ACTIONS
jerry-smartloan-dep	jerry	jerry-smartloan v1	mlz16	online	PMML	Feb 5, 2019 11:52 AM	—	<span style="color: red;">⋮</span>

1 deployment in total

Click the “**View details**” button under “**ACTIONS**” to check the details of the deployment.

The screenshot shows the same 'Deployments' page as above, but with a context menu open over the first row. The 'View details' option in the menu is highlighted with a red box.

- Schedule evaluation
- Test API
- Update
- Delete

Note the API details for the RESTful API provided by the scoring service of WMLz which allows your application to call the model for online scoring. In the next step, you will test this scoring API and the model online in the Web UI. If the test is successful, you can then have your application developer update the application to call this REST API. This allows you to implement online scoring on the same system as the transactions running under z/OS.

IBM Watson Machine Learning for z/OS

Deployments > jerry-smartloan-dep - Overview

DEPLOYMENT NAME: jerry-smartloan-dep

SCORING ENDPOINT: https://9.30.128.39:30163/ml/v2/scoring/online/cfc1aed4-ba8a-4185-bd85-0c961fabe380

ONLINE FEEDBACK ENDPOINT: —

API SPECIFICATION: API specification for scoring endpoint is available. [Download](#)

PUBLISHER: jerry

DATE DEPLOYED: Feb 5, 2019 11:52 AM

ASSOCIATED MODEL NAME: jerry-smartloan [vi](#)

TYPE: online

ENGINE: PMML

NEXT EVALUATION TIME: —

SCORING SERVICE: mlz16 (9.30.128.39:30163)

NUMBER OF INVOCATIONS: 0

AVERAGE ELAPSED TIME: 0 ms

REQUEST HEADER: See the API documentation for more details

**Model Schema**

Input Schema			Output Schema		
NAME	TYPE	NULLABLE	NAME	TYPE	NULLABLE
ds	real	false	R-loan_status	string	false
home_ownership	string	false	RC-loan_status	real	false
term	real	false	Rloan_status	string	false
total_current_balance	real	false	RP-charged_off	real	false
			RP-fully_paid	real	false

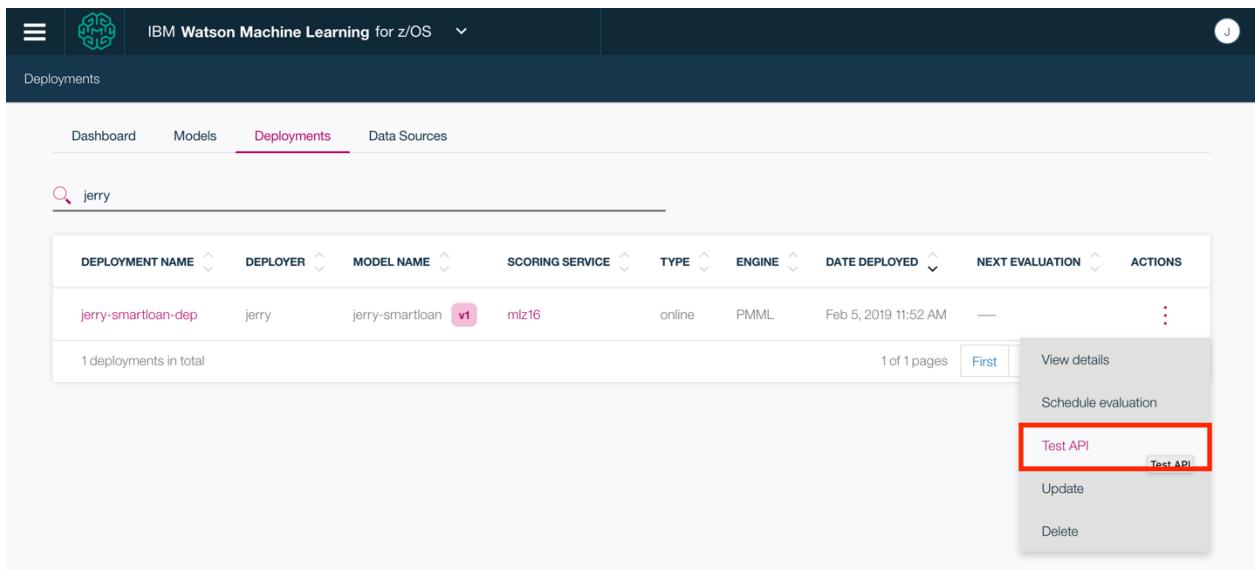
[Table](#) [JSON](#) [JSON Schema](#)

**Evaluation Results**

START TIME	END TIME	STATUS	RECORDS COUNT
no results found			

[Schedule evaluation](#)

- 10. Go back to the “**Deployments**” page, click the “**Test API**” button under “**ACTIONS**”



The screenshot shows the IBM Watson Machine Learning for z/OS interface. The top navigation bar includes a menu icon, the product logo, and the text "IBM Watson Machine Learning for z/OS". Below the header, the word "Deployments" is highlighted in blue. The main content area has tabs for "Dashboard", "Models", "Deployments" (which is active), and "Data Sources". A search bar contains the text "jerry". The main table lists one deployment:

DEPLOYMENT NAME	DEPLOYER	MODEL NAME	SCORING SERVICE	TYPE	ENGINE	DATE DEPLOYED	NEXT EVALUATION	ACTIONS
jerry-smartloan-dep	jerry	jerry-smartloan v1	mlz16	online	PMML	Feb 5, 2019 11:52 AM	—	<a href="#">View details</a> <a href="#">Schedule evaluation</a> <a href="#">Test API</a> <a href="#">Update</a> <a href="#">Delete</a>

Below the table, it says "1 deployments in total". On the right side of the table, there are buttons for "View details", "Schedule evaluation", "Test API" (which is highlighted with a red box), "Update", and "Delete".

- 11. On the “**Test API**” page, enter a record for values according to the schema of the input record of the model as below. Input values are case sensitive!

IBM Watson Machine Learning for z/OS

Deployments > Deployment: dddd > Test API

**Input**

Table      JSON

**RATEPLAN \***  
3

**GENDER \***  
M

**ESTINCOME \***  
400

**STATUS \***  
M

**USAGE \***  
60

**CHILDREN \***

**Clear**      **Submit**

### Sample Input

rateplan:  
gender:  
estincome:  
status:  
usage:  
children:  
age:  
dropped:  
localbilltype:  
longdistance:  
local:  
international:  
longdistancebillt:  
paymethod:  
carowner:

- 12. Click the “**Submit**” button to test the prediction. The result will be shown in “**Result**” area.

IBM Watson Machine Learning for z/OS

Deployments > Deployment: dddd > Test API

**Input**

Table JSON

**RATEPLAN \***  
3

**GENDER \***  
F

**ESTINCOME \***  
30000

**STATUS \***  
S

**USAGE \***  
60

**CHILDREN \***

Clear Submit

**Result**

```
[  
 {  
 "prediction": "F",  
 "probability": [  
 0.6,  
 0.4  
 ]  
 }  
]
```

## We Value Your Feedback!

Don't forget to submit your session feedback! Your feedback is very important to us – we use it to continually improve the lab.