Lab Guide

CP4Data and Z Integration HOL

Session: EZS66T316BH

Maggie Lin meichi@us.ibm.com, Watson Machine Learning for z/OS Development

Table of Contents

Legal Disclaimer

 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 dam ages 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.

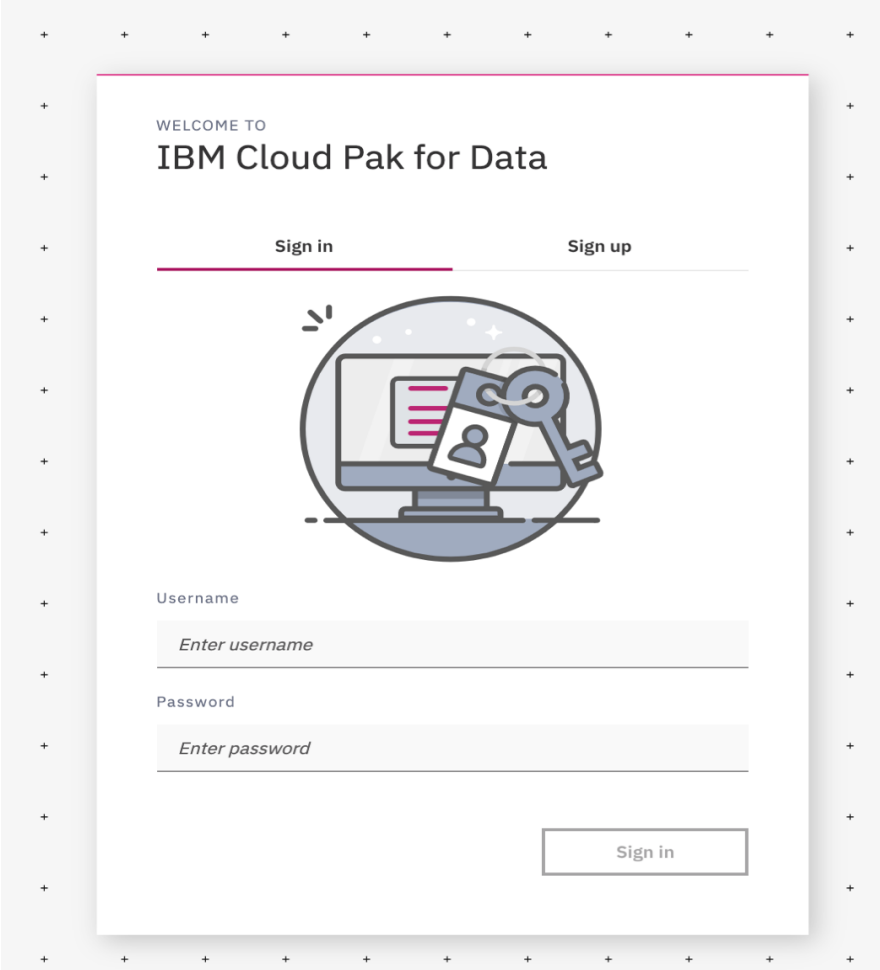
- Model Training with Jupyter notebook
- Model Management and Deployment

Log in to CP4D

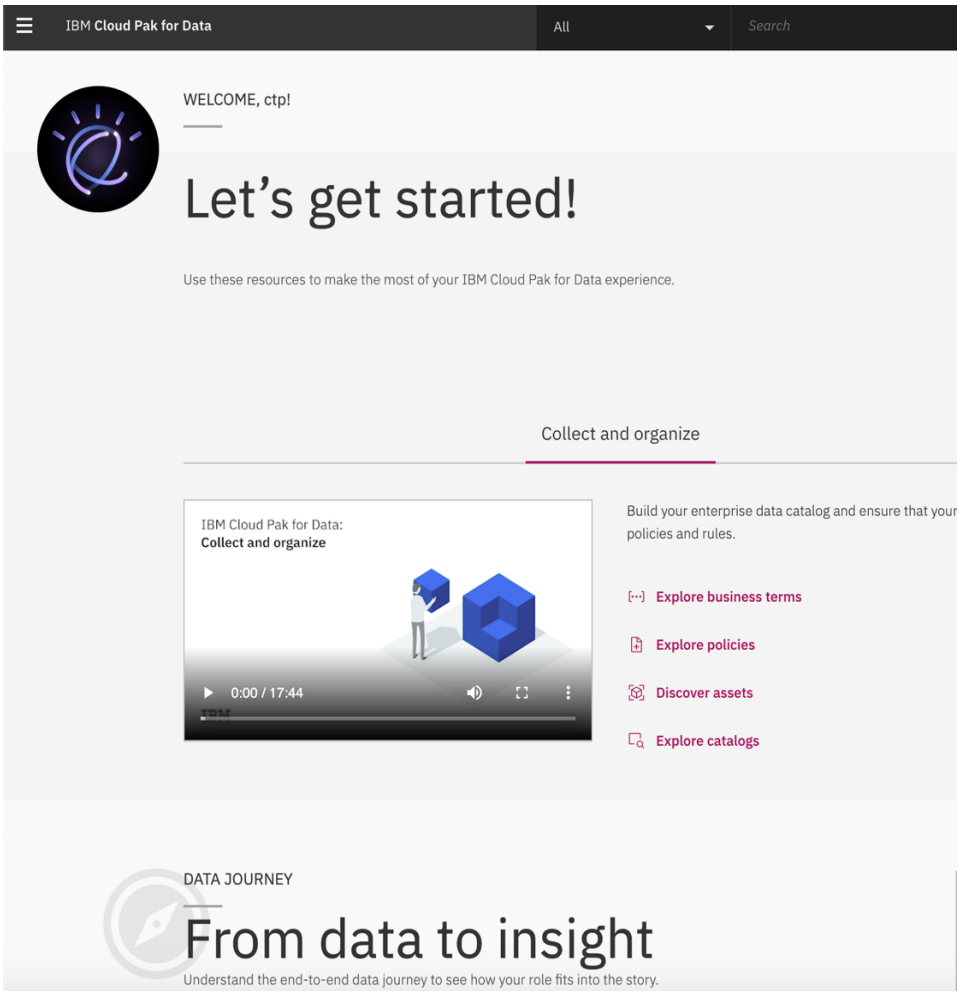
The URL and authentication credentials for the CP4D Web UI are provided on a separate worksheet. Contact the lab instructor if you did not receive a worksheet.

__ 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

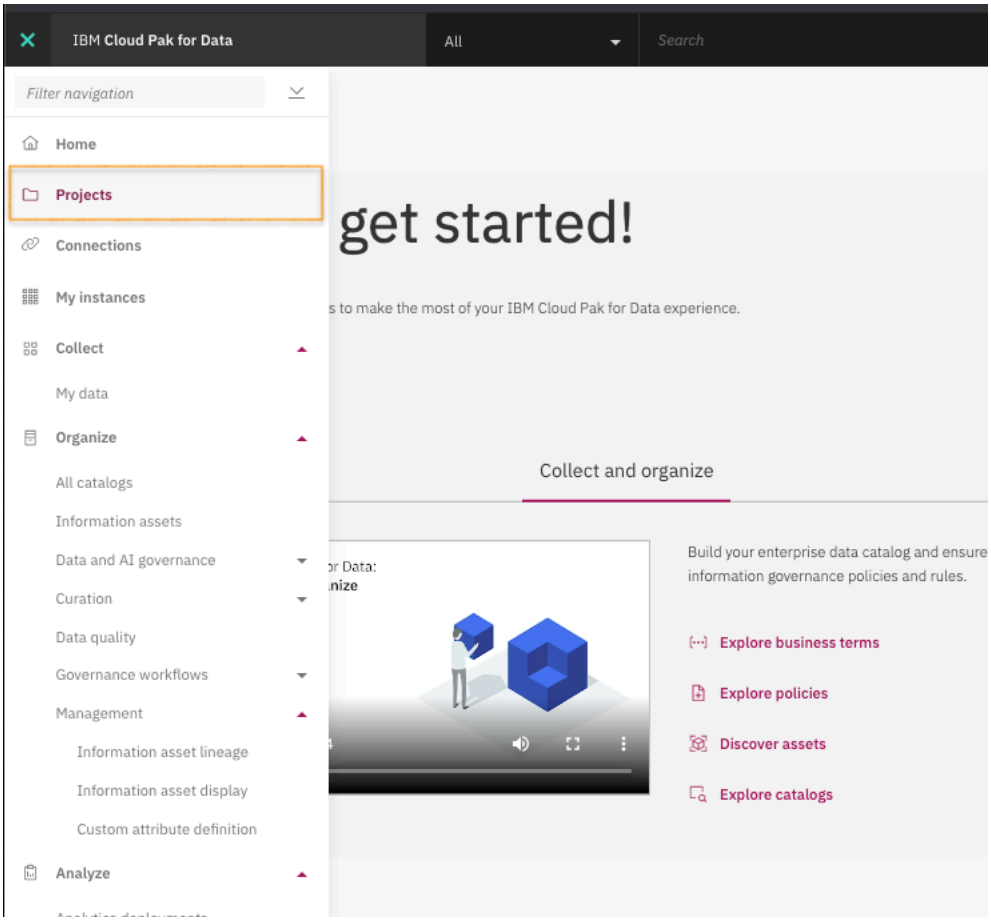


1. Model Training in CP4D

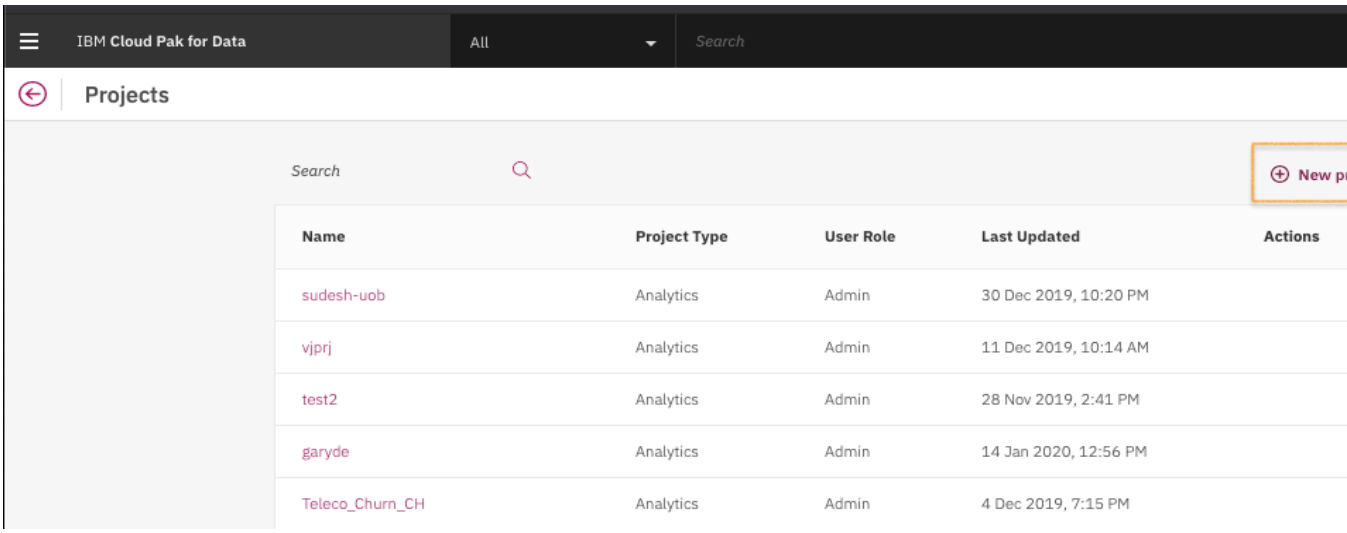
In this Lab, we will use the Web UI of CP4D to leverage machine learning to understand customer churn and determine which customer is likely to leave. You will learn how to use Jupyter notebook of CP4D to train a binary classification scikit learn (RandomForestClassifier) model.

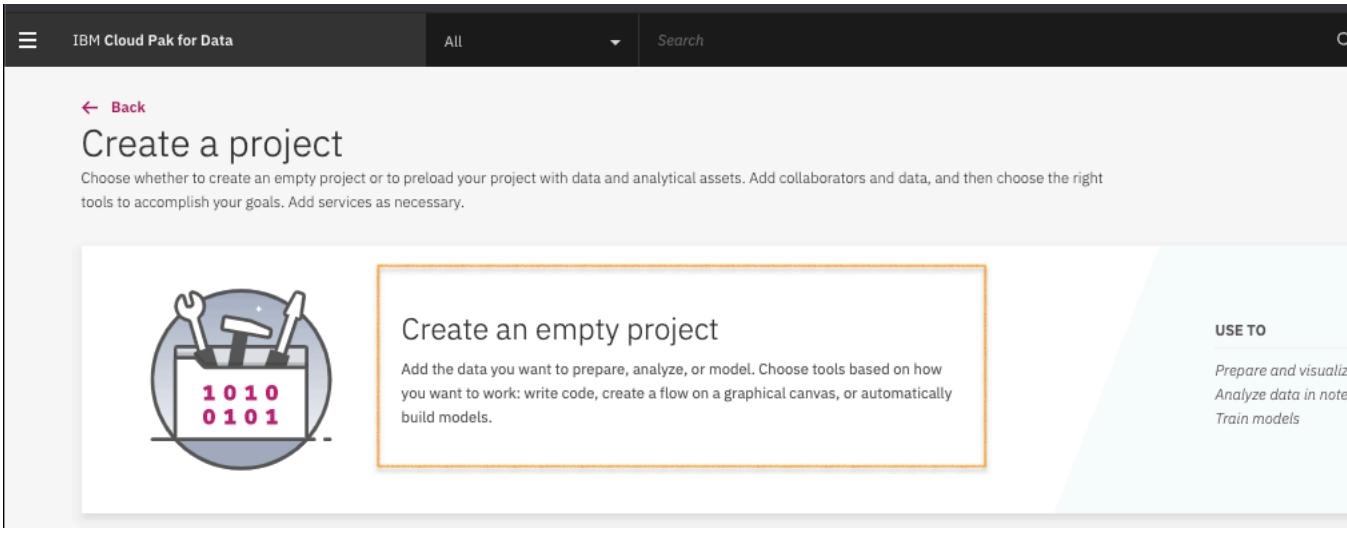
__ 1. Select a project to contain your notebook.

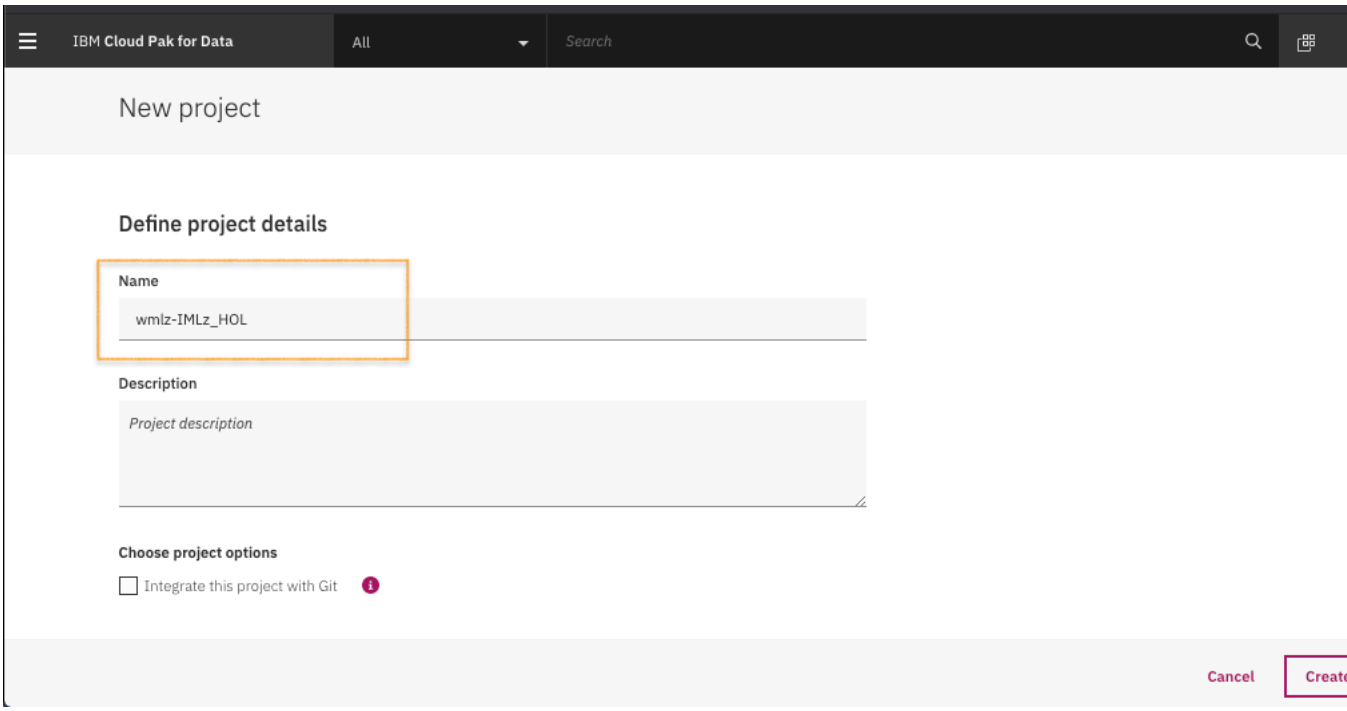
Click on the 3 short horizontal bars in the top left of the web page and choose "**Projects**"



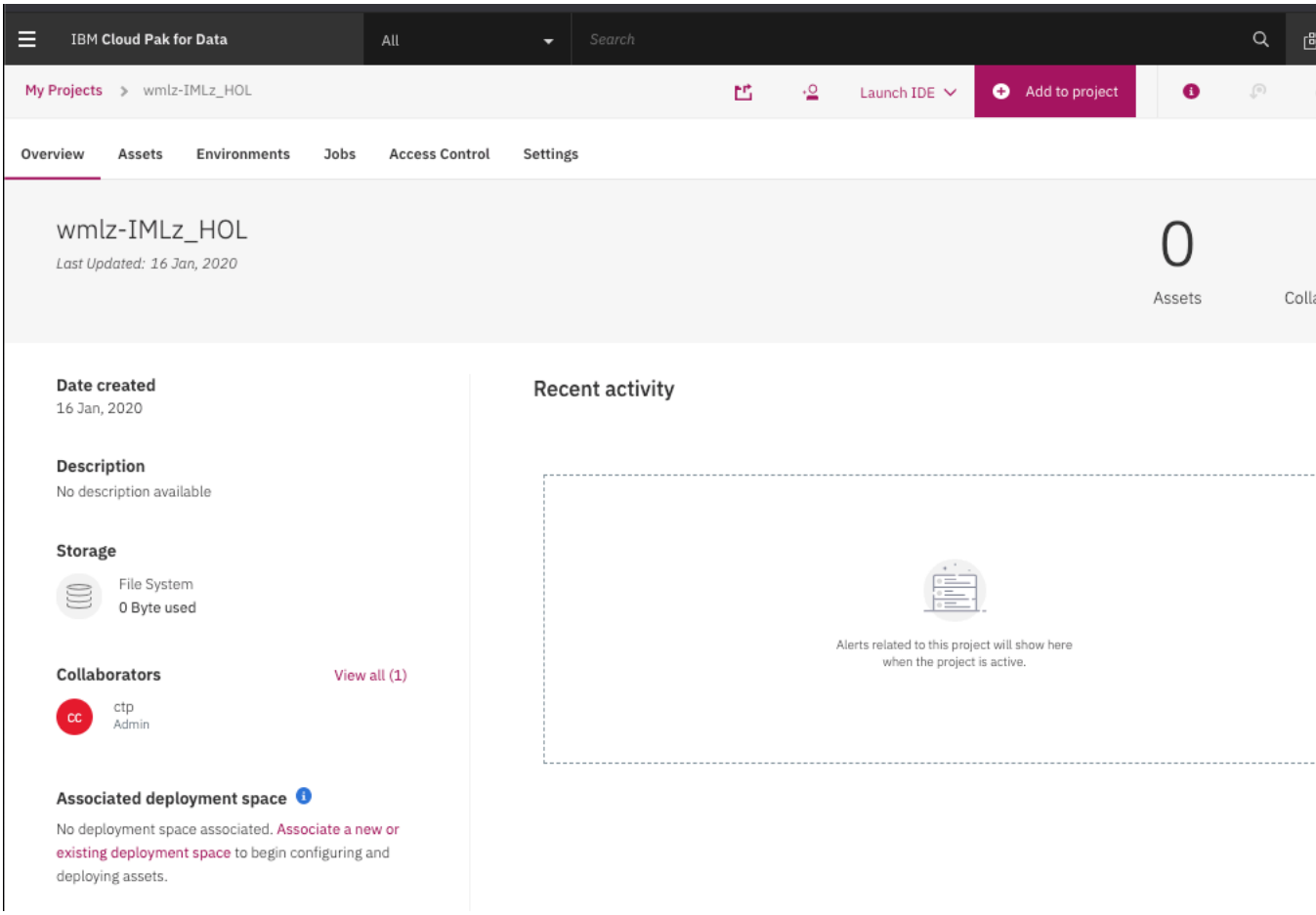
Click "**New Project**" button to create one. Select to create an empty project, and name it "**<cp4d_login_userid>-IMLz_HOL**



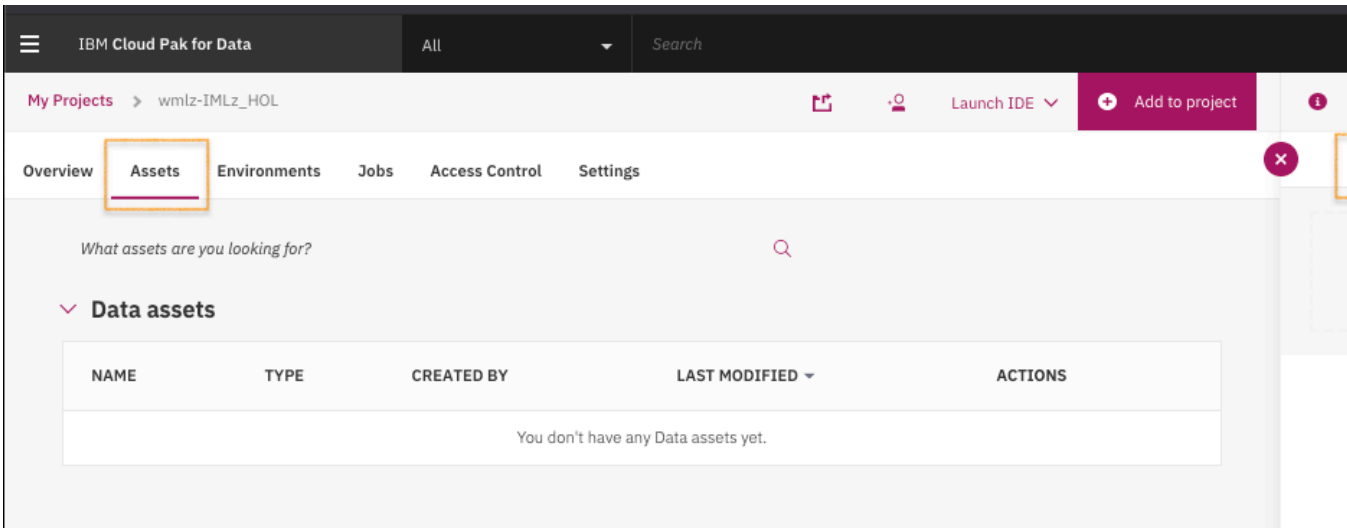


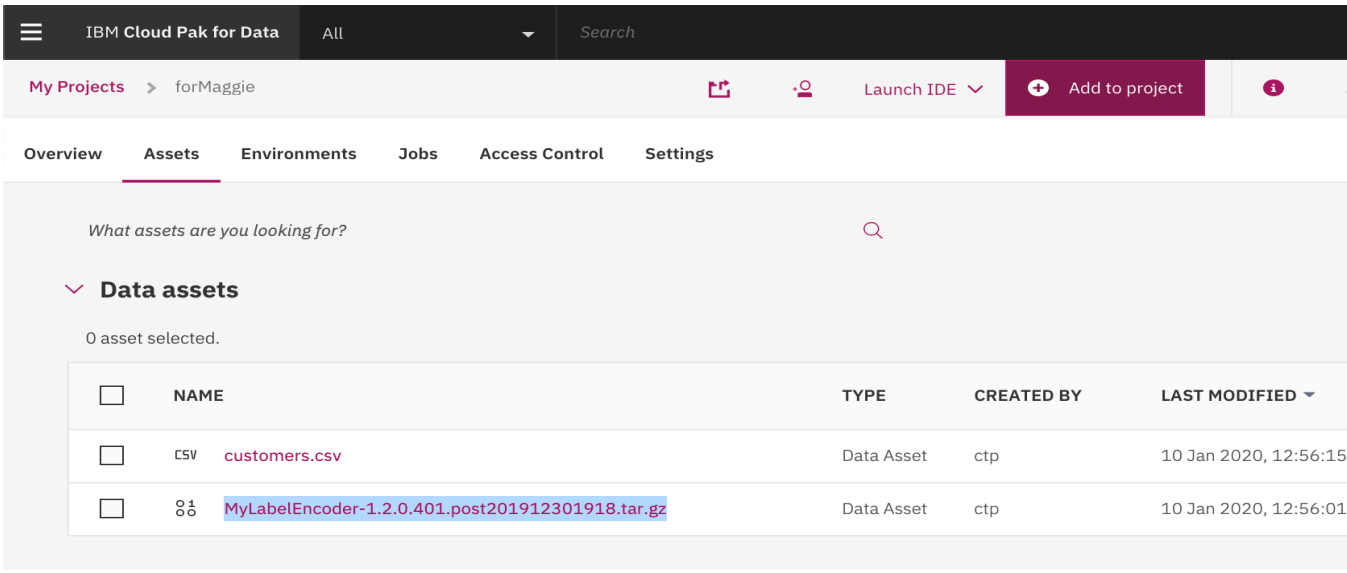


You should be now at the project overview page.

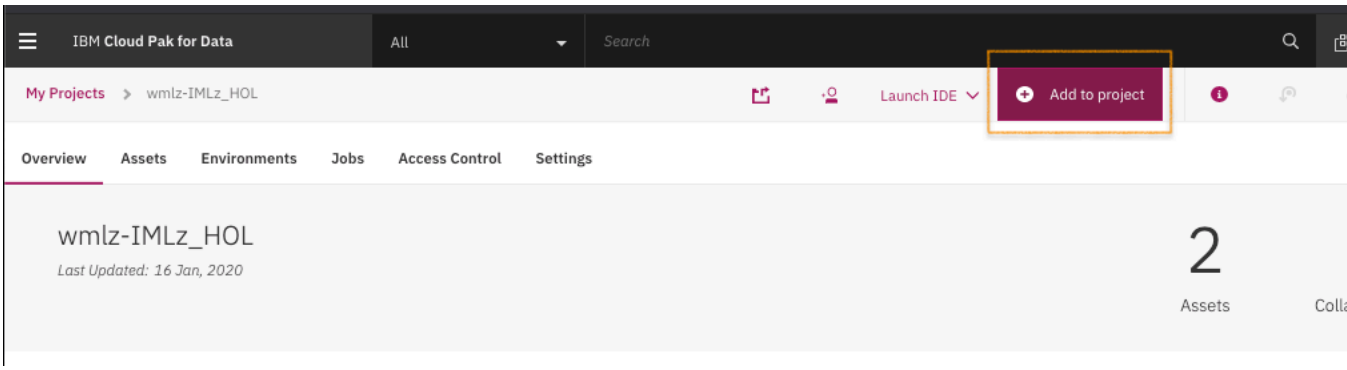


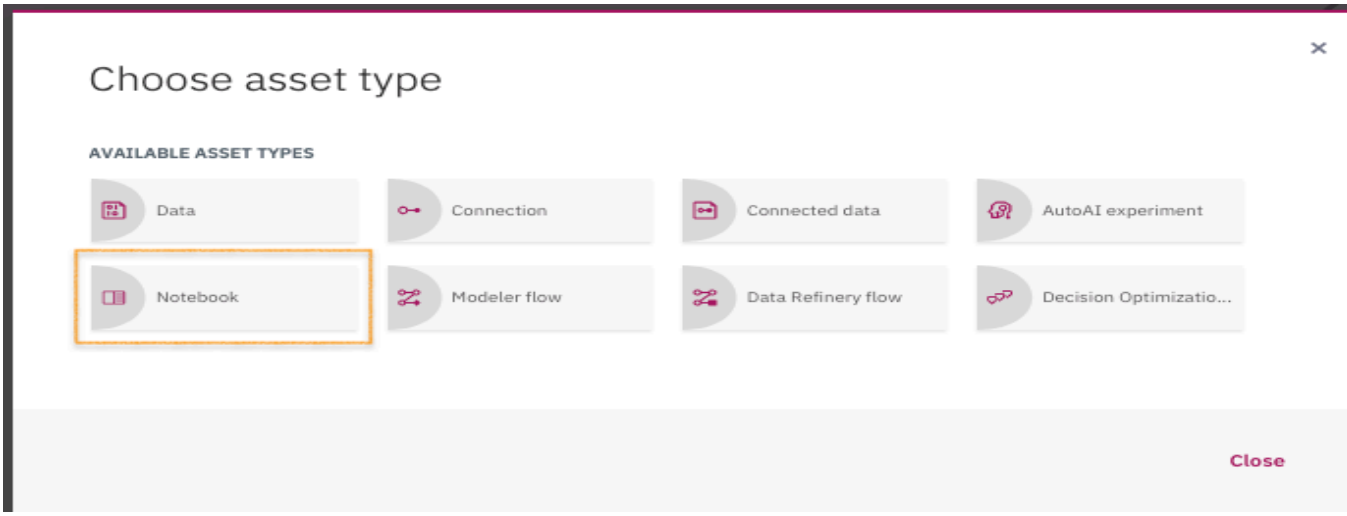
__ 2. From the project overview, click on “Asset” to load 2 asset files **customers.csv** and **MyLabelEncoder-1.2.0.401.post201912301918.tar.gz**
Download the 2 files from the HOL GitHub link if you have not :
https://github.com/meichilin/IBMWMLz/tree/master/FastStart_2020



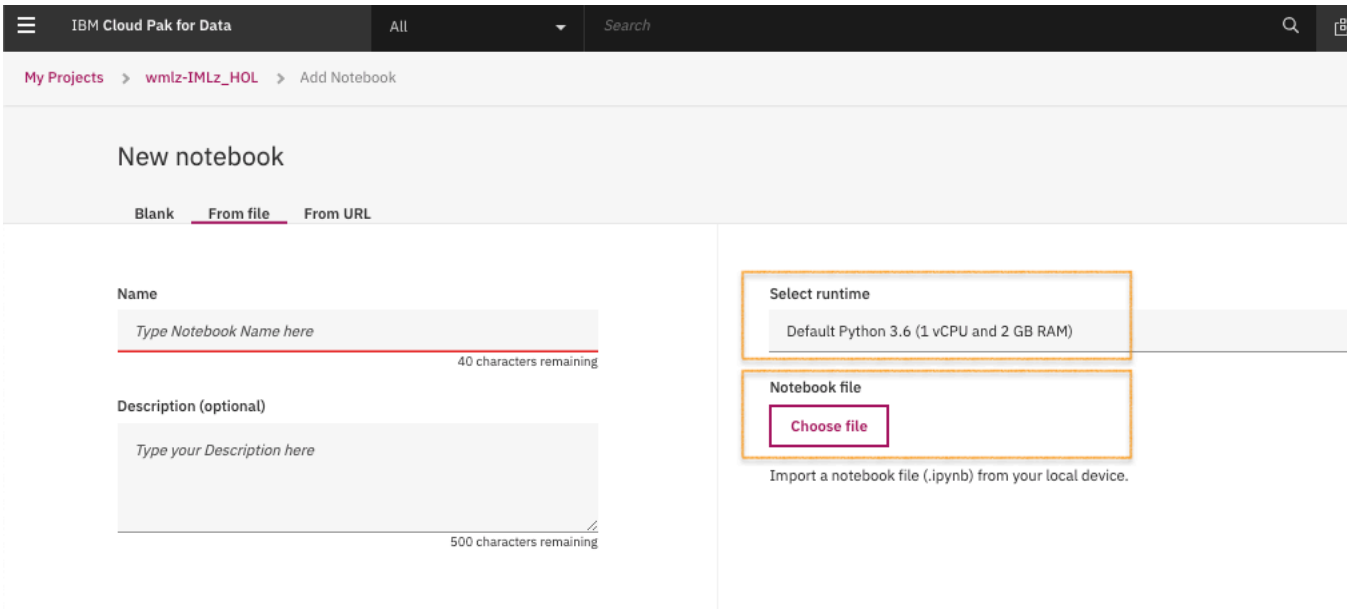


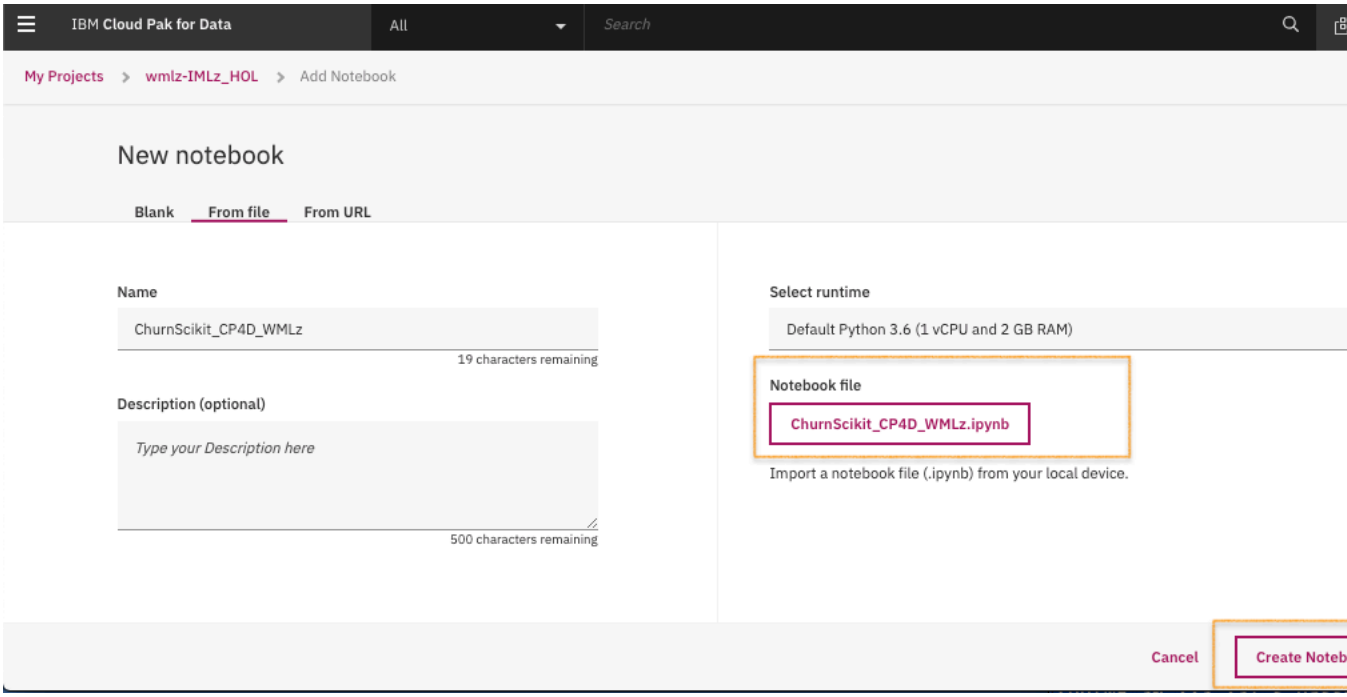
__ 3. Click “Add to project” to add “Notebook” asset to the project.





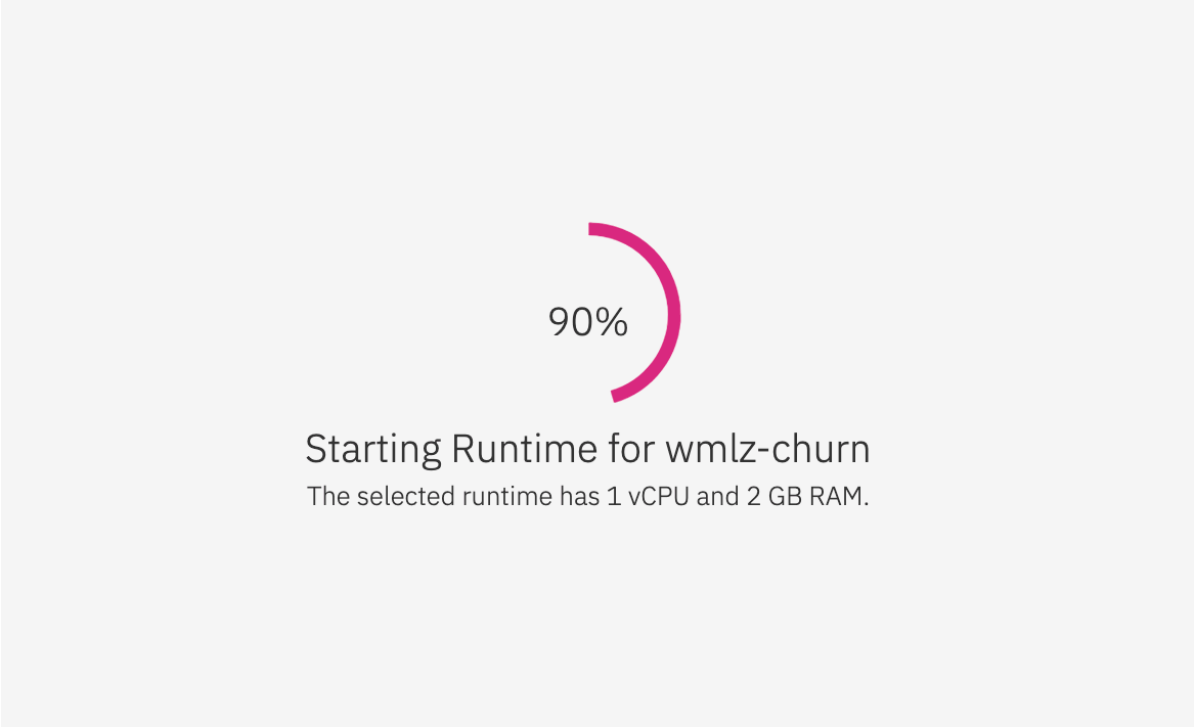
__ 4. Create a new notebook “From file”. Import **ChurnScikit_CP4D_WMLz.ipynb** from the HOL GitHub link if you have not : https://github.com/meichilin/IBMWMLz/tree/master/FastStart_2020. Leave runtime to Default Python 3.6 (1 vCPU and 2 GB RAM). Then click “**Create Notebook**” button.



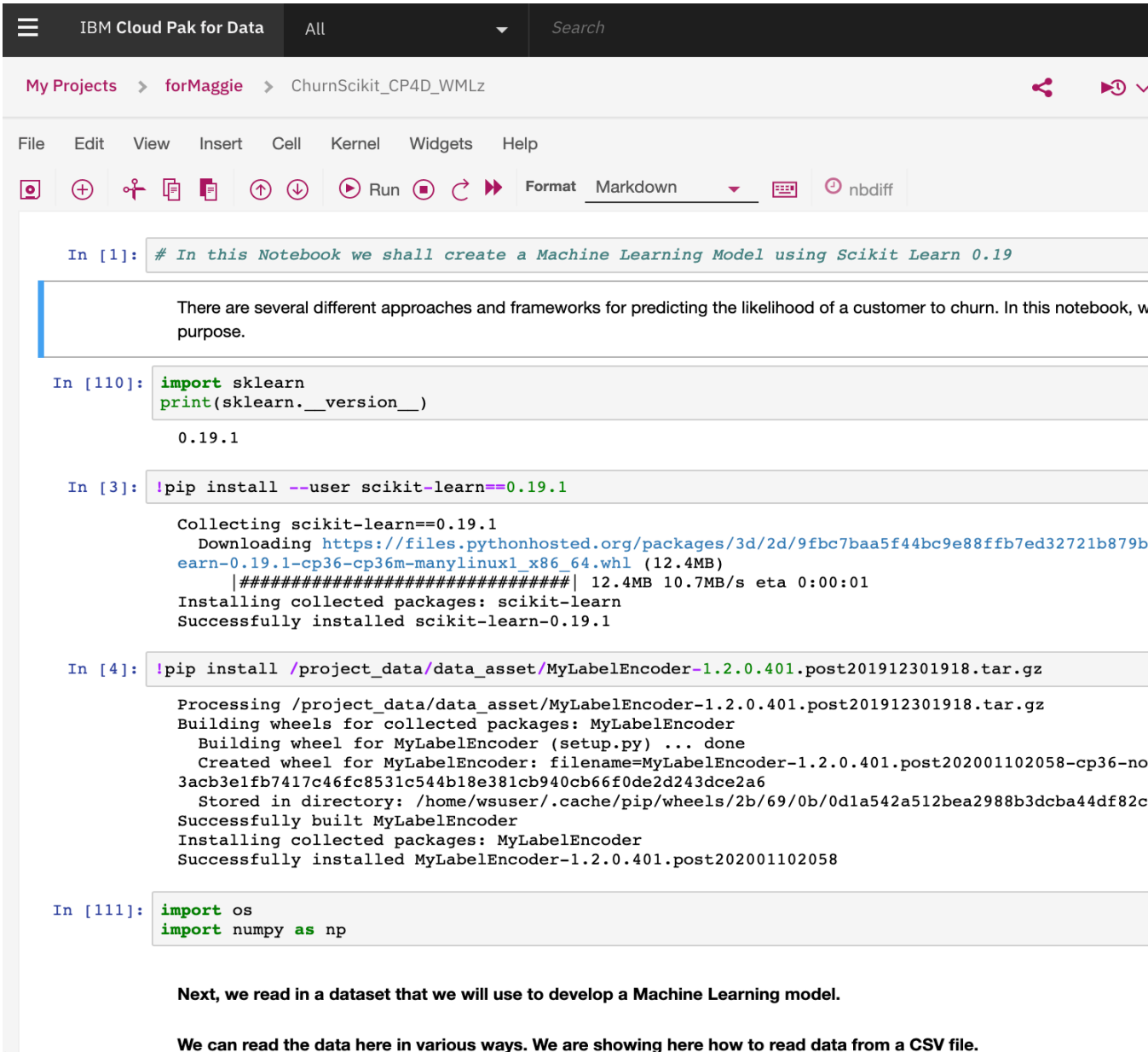


You will see the following spinner indicating a runtime container is being created for the Jyputer notebook.

This could take a minute or two for the first time you open a notebook container in your project.



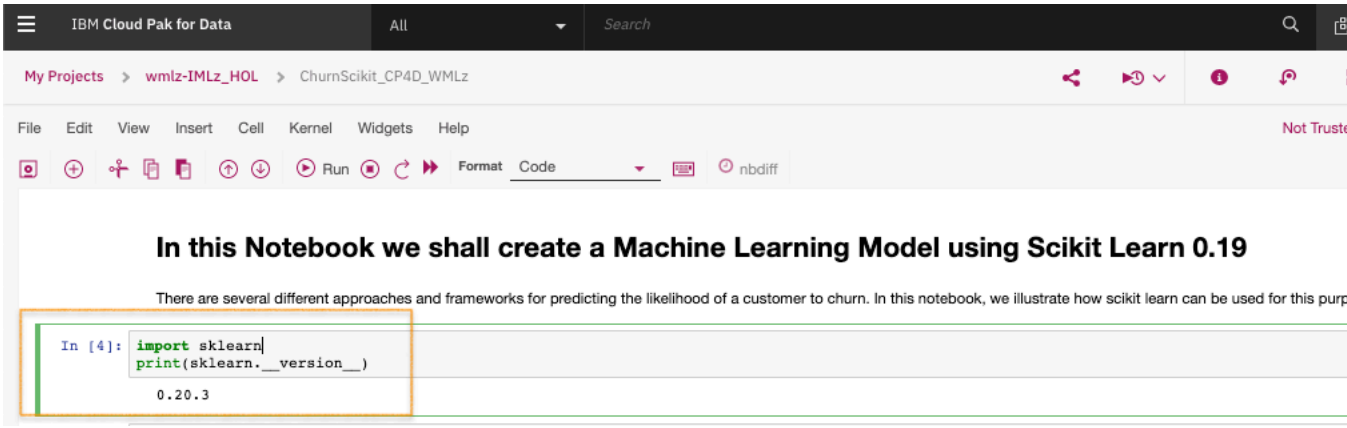
A message indicates the Jupyter notebook is being launched. Shortly afterward, you will see the ChurnScikit_CP4D_WMLz.ipynb loaded.



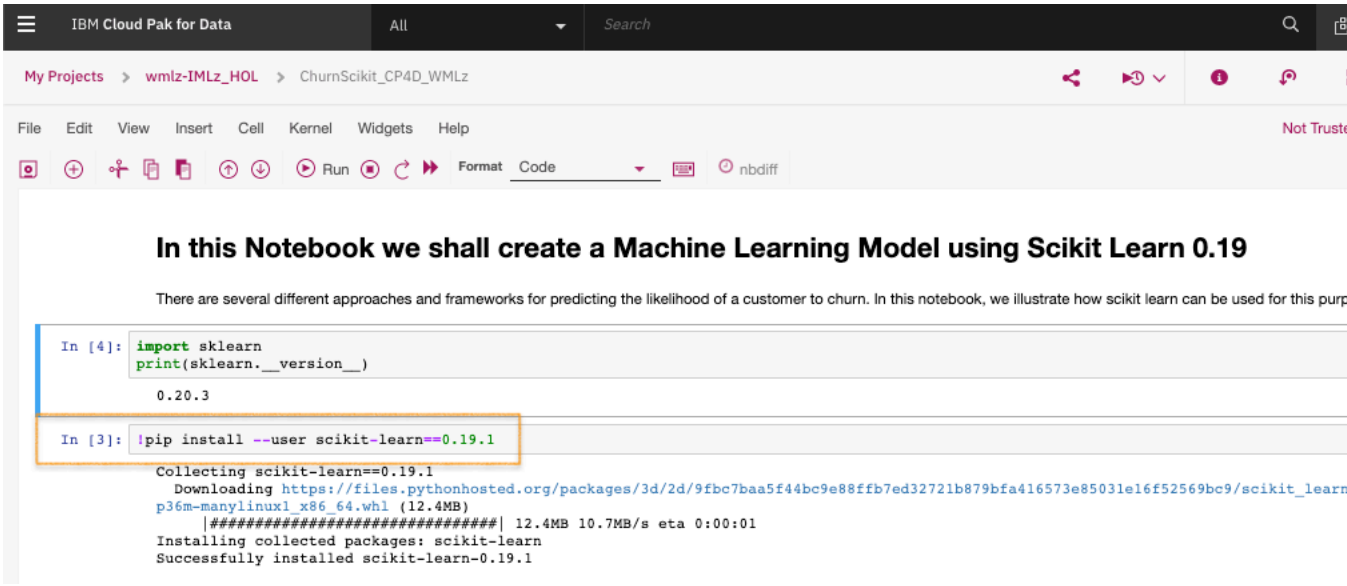
__ 5. Now to train the model, execute each notebook cell step by step.

First, make sure we are using python scikit learn package 0.19.1 version, and install the customized transformer **MyLabelEncoder.**

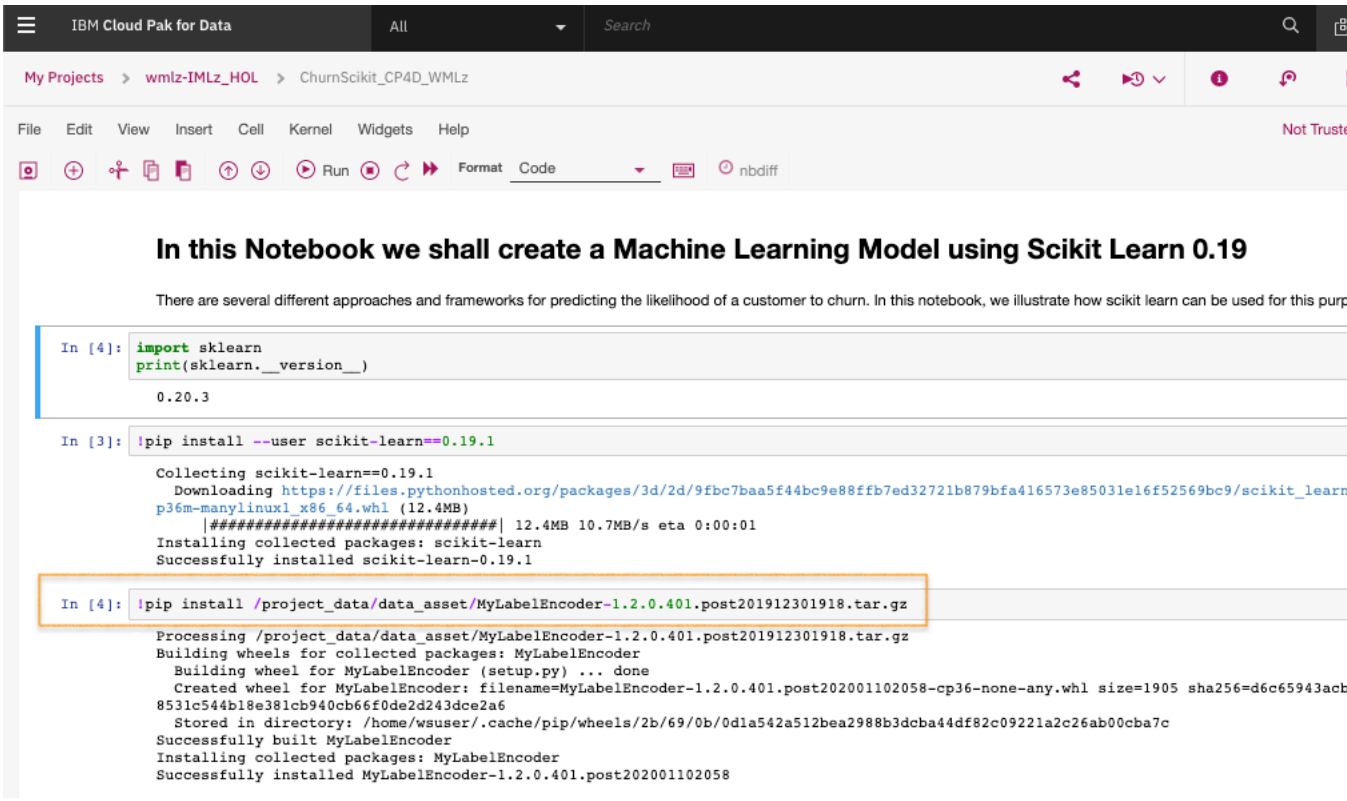
Execute the first cell to know the current scikit learn package version. If it is already at 0.19.1 version, you can proceed on to the next step to install the customized transformer **MyLabelEncoder.**



Pip install the scikit learn package to 0.19.1 version.



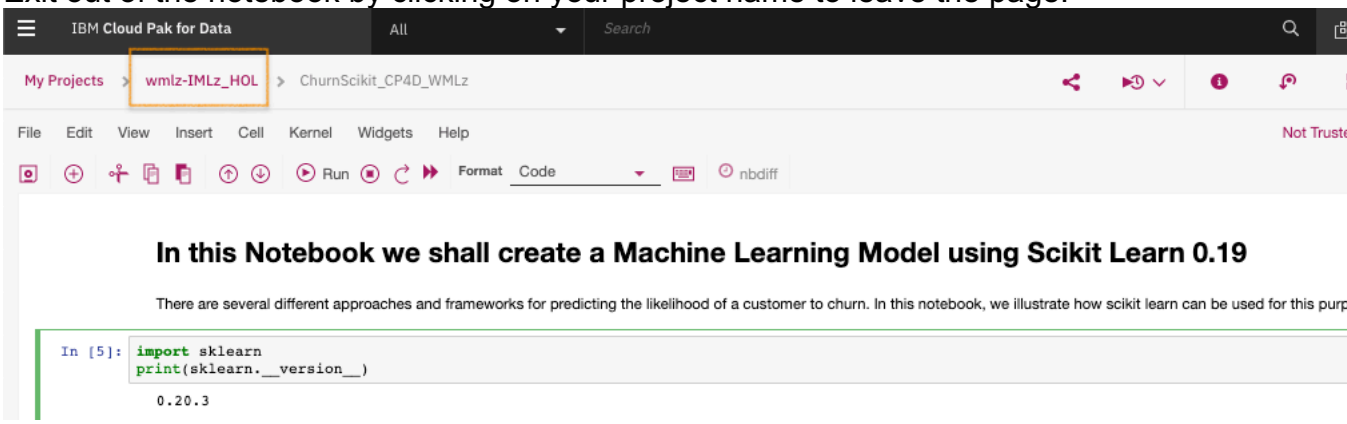
Pip install the customized transformer **MyLabelEncoder**.



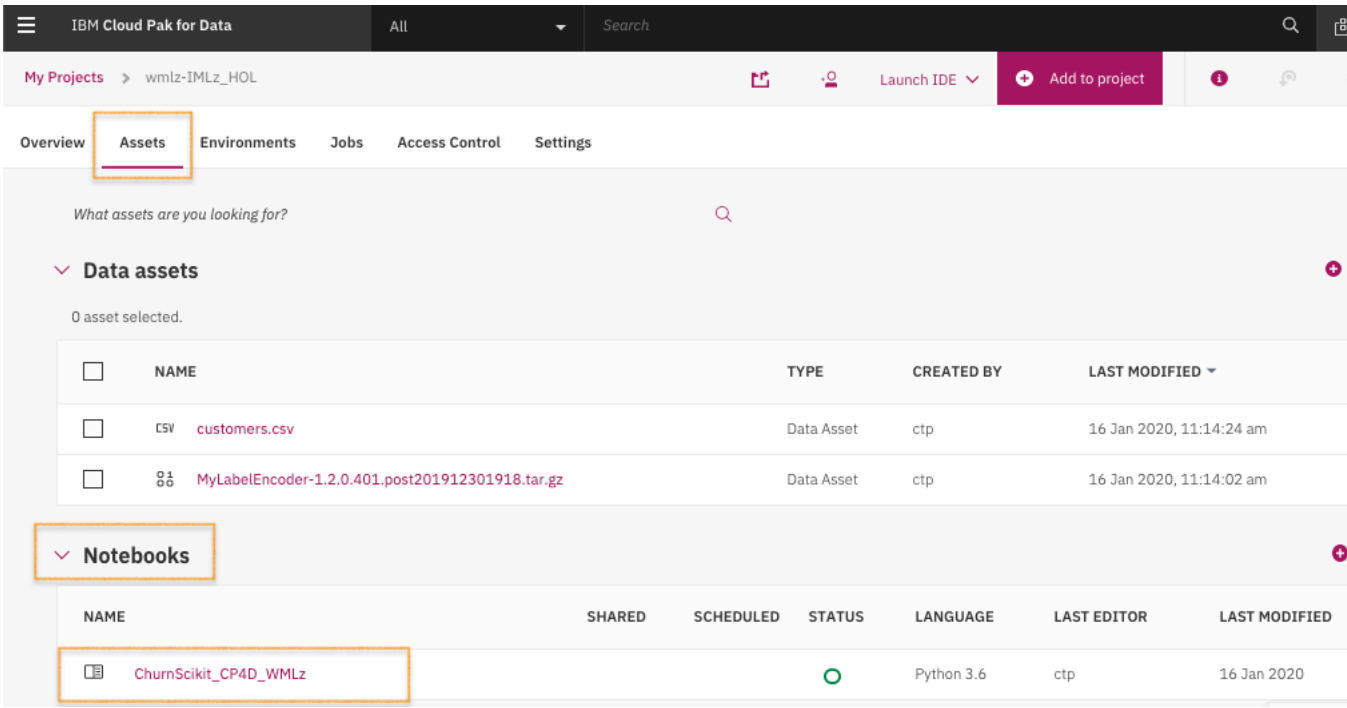
__ 6. If your scikit learn package version is already in 0.19.1 version, skip this step.

Otherwise, exit the notebook to restart the kernel to let the new the scikit learn package version to take effect in the notebook session.

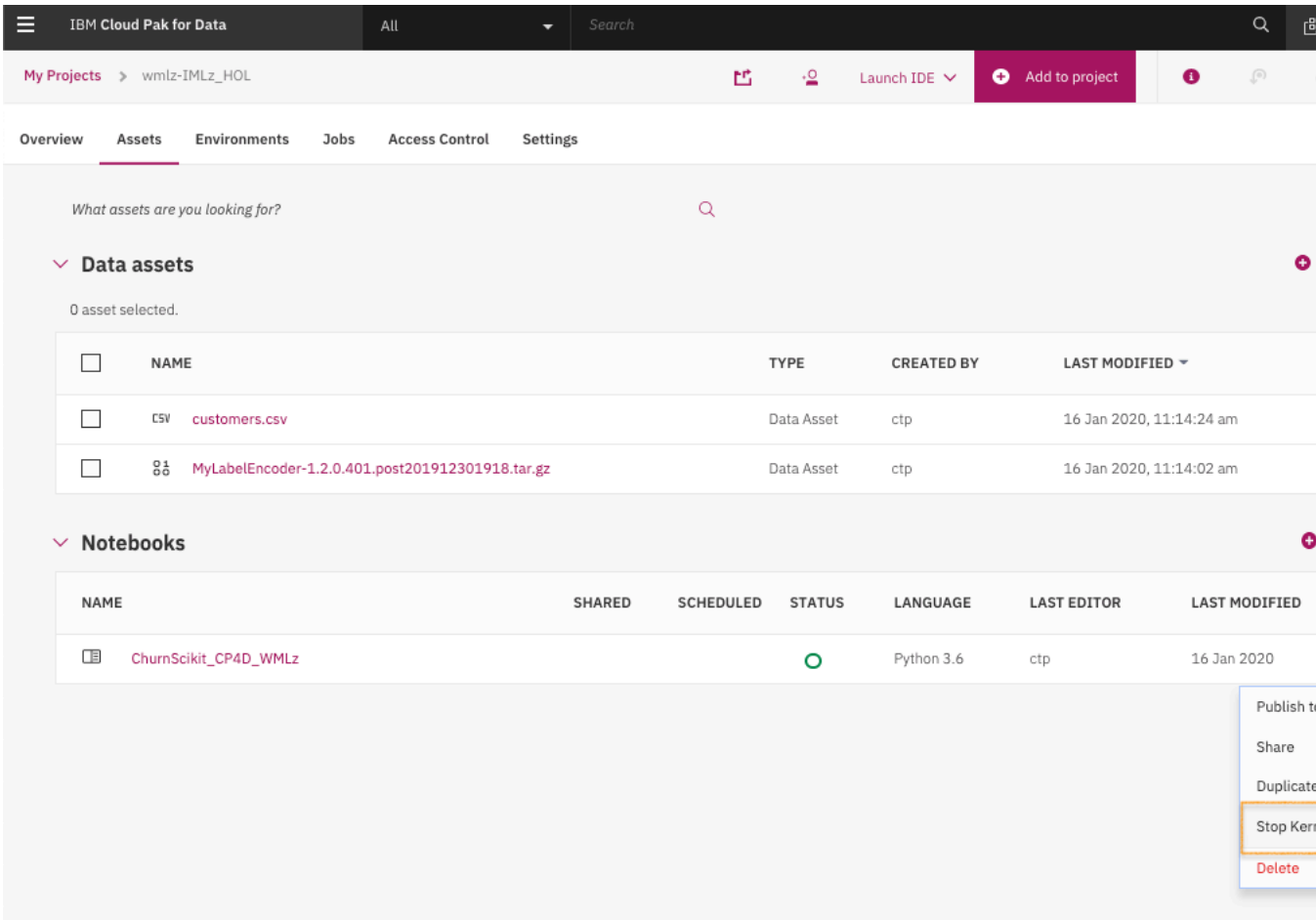
Exit out of the notebook by clicking on your project name to leave the page.



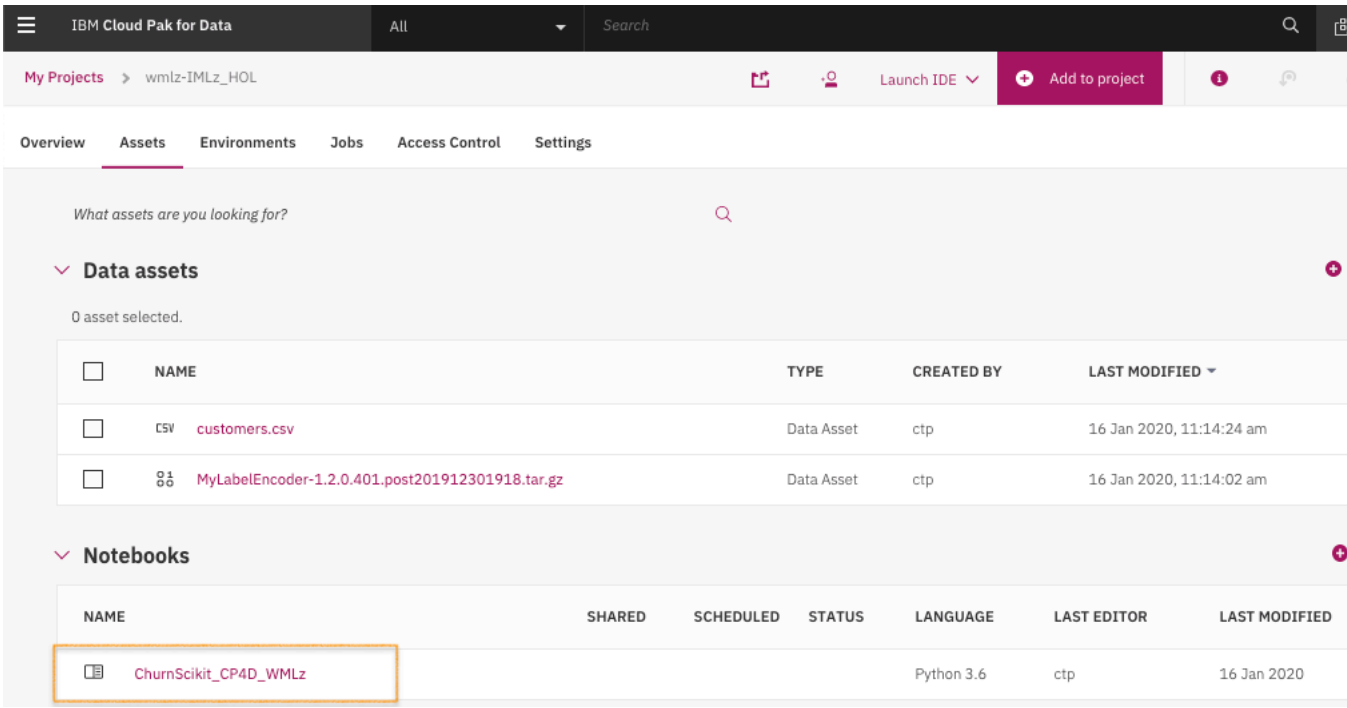
In **Assests** page, **Notebooks** section, locate the ChurnScikit_CP4D_WMLz notebook.



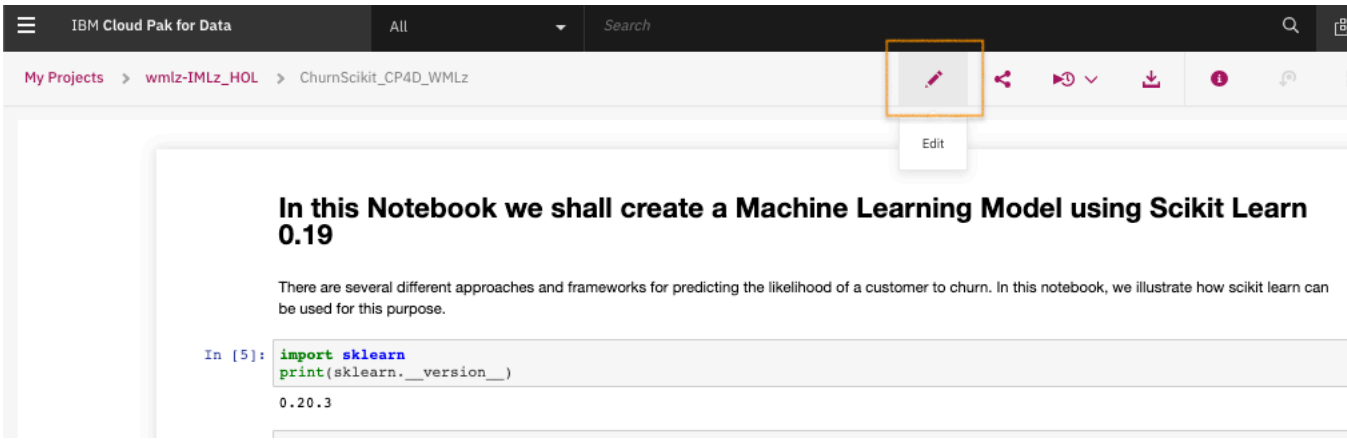
Under **Action,** click on it to **Stop Kernel.**



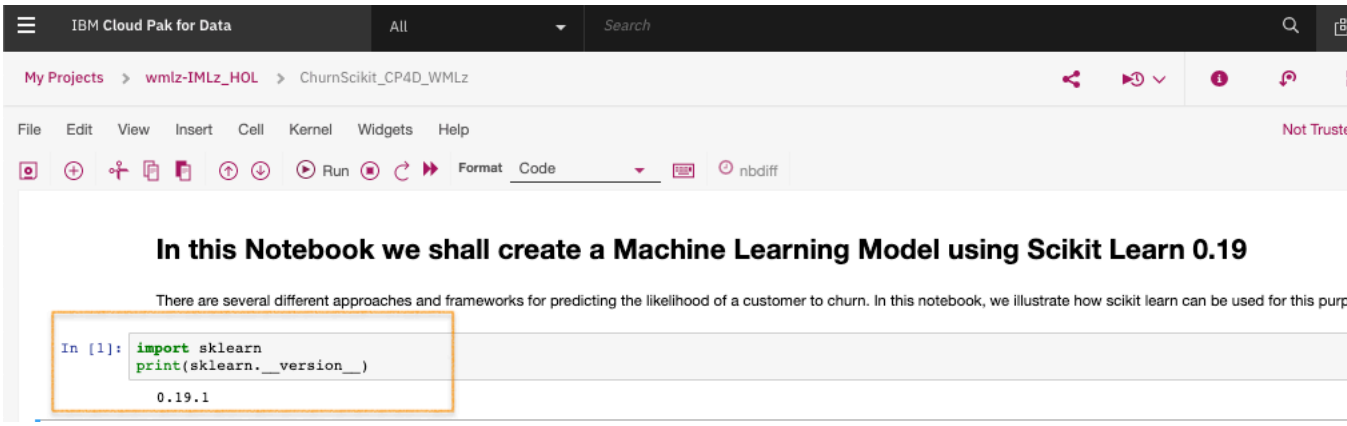
Once the kernel is stopped, go back into the **ChurnScikit_CP4D_WMLz** notebook to continue.



Enable the notebook in edit mode for continous notebook execution.



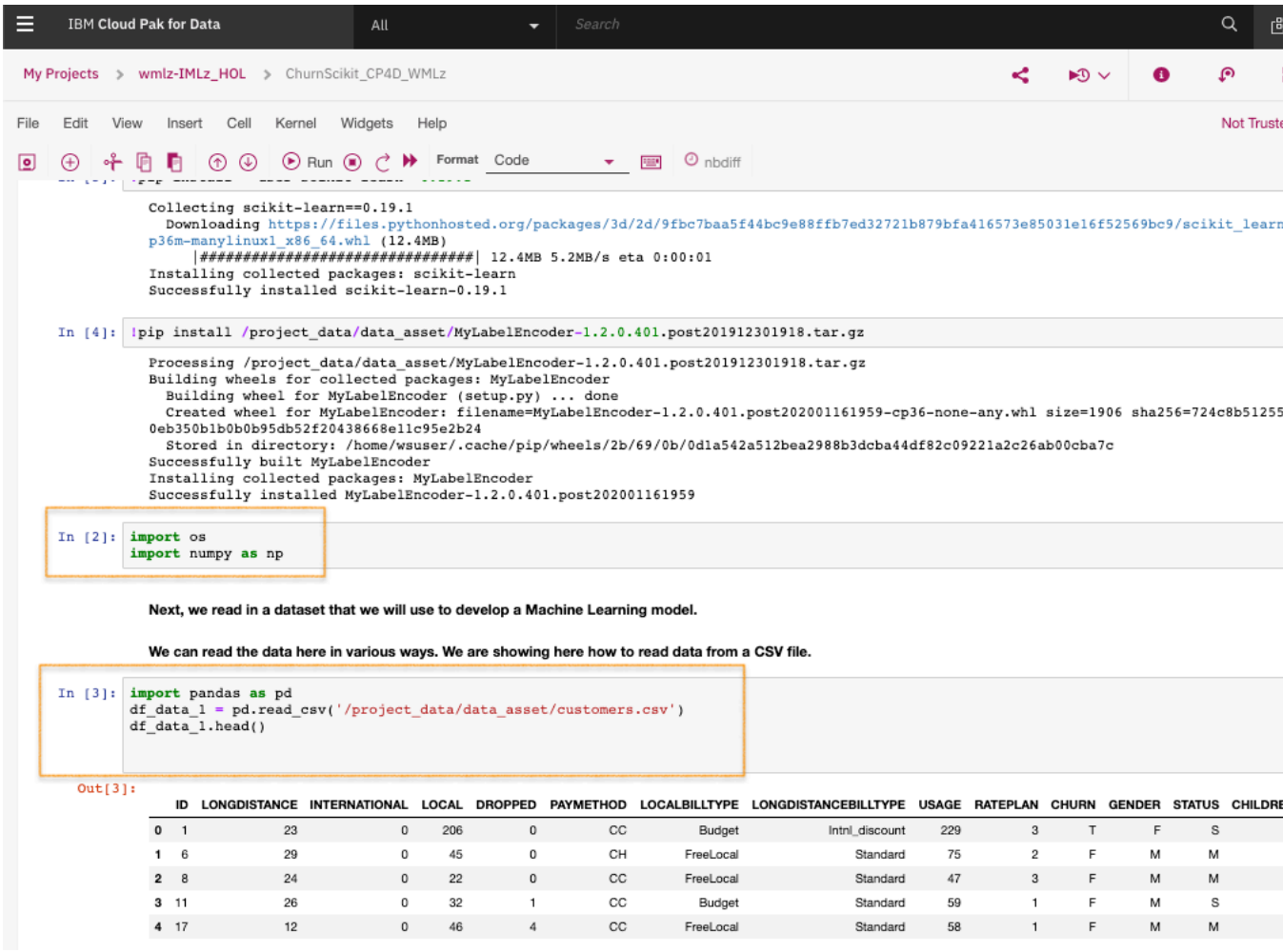
Lets make sure the python scikit learn package shows 0.19.1 version.



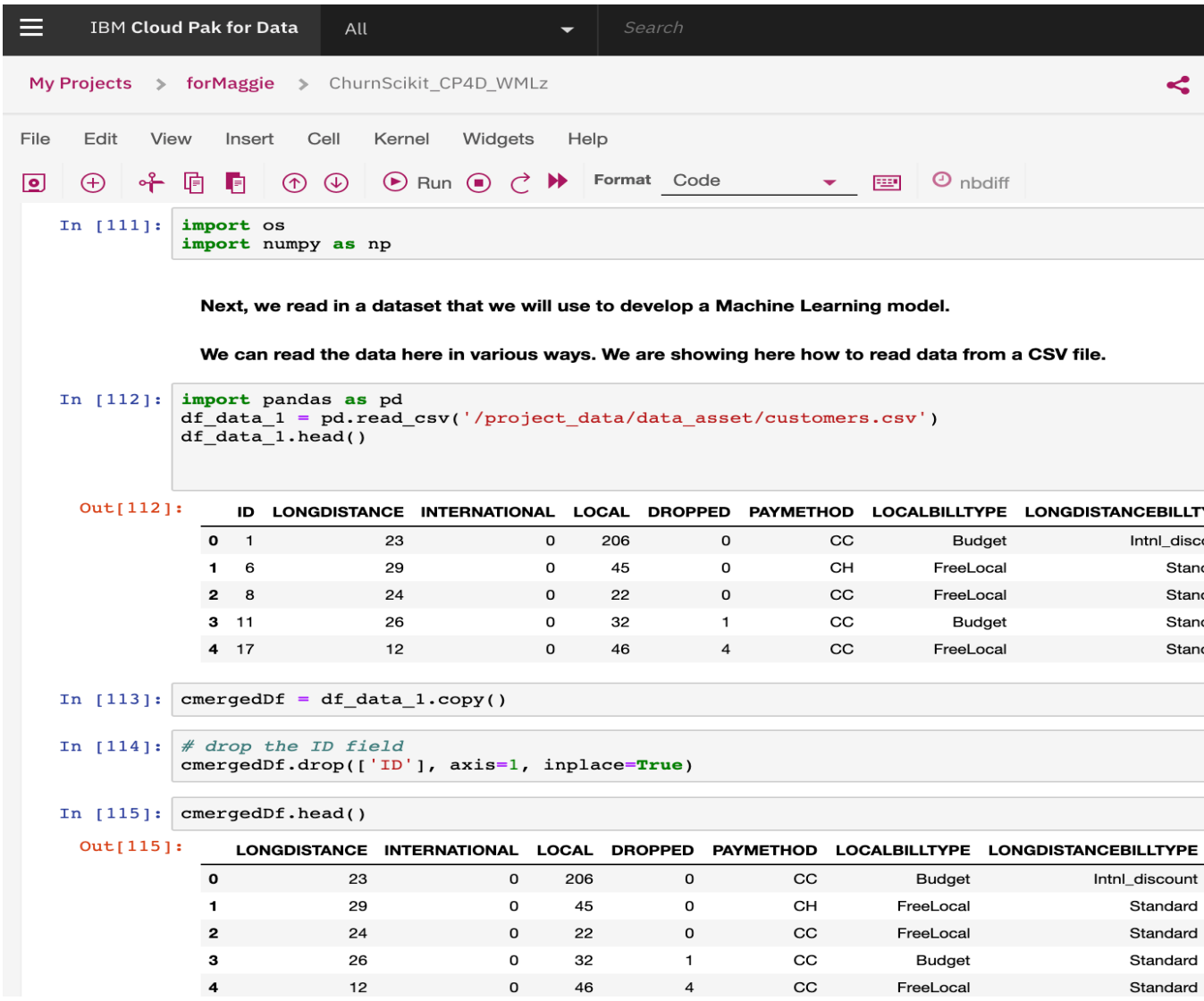
__ 7. Lets continue on to execute the notebook cell ***one by one***.

Import numpy package.

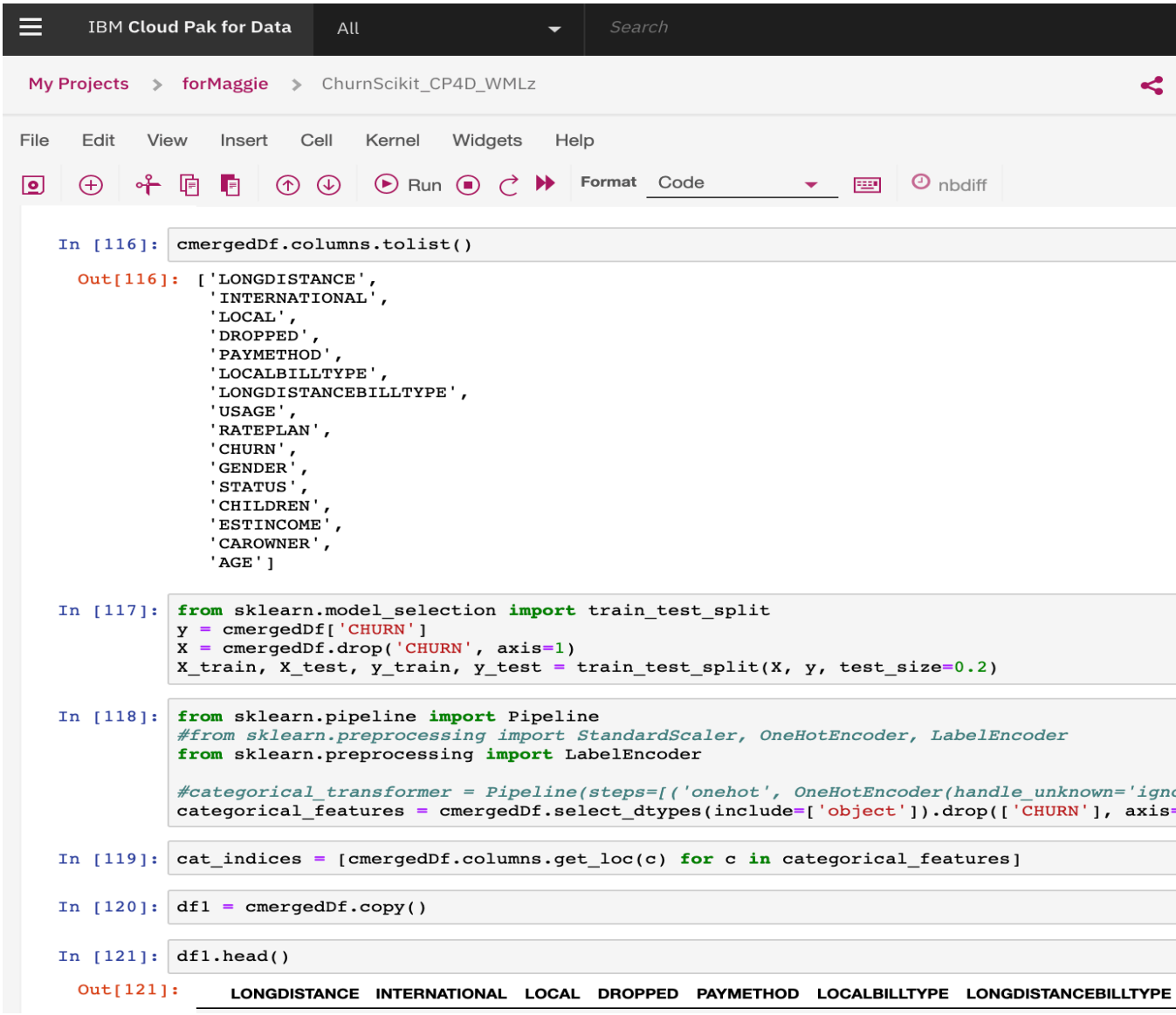
Load the training data **customer.csv** and split training data.



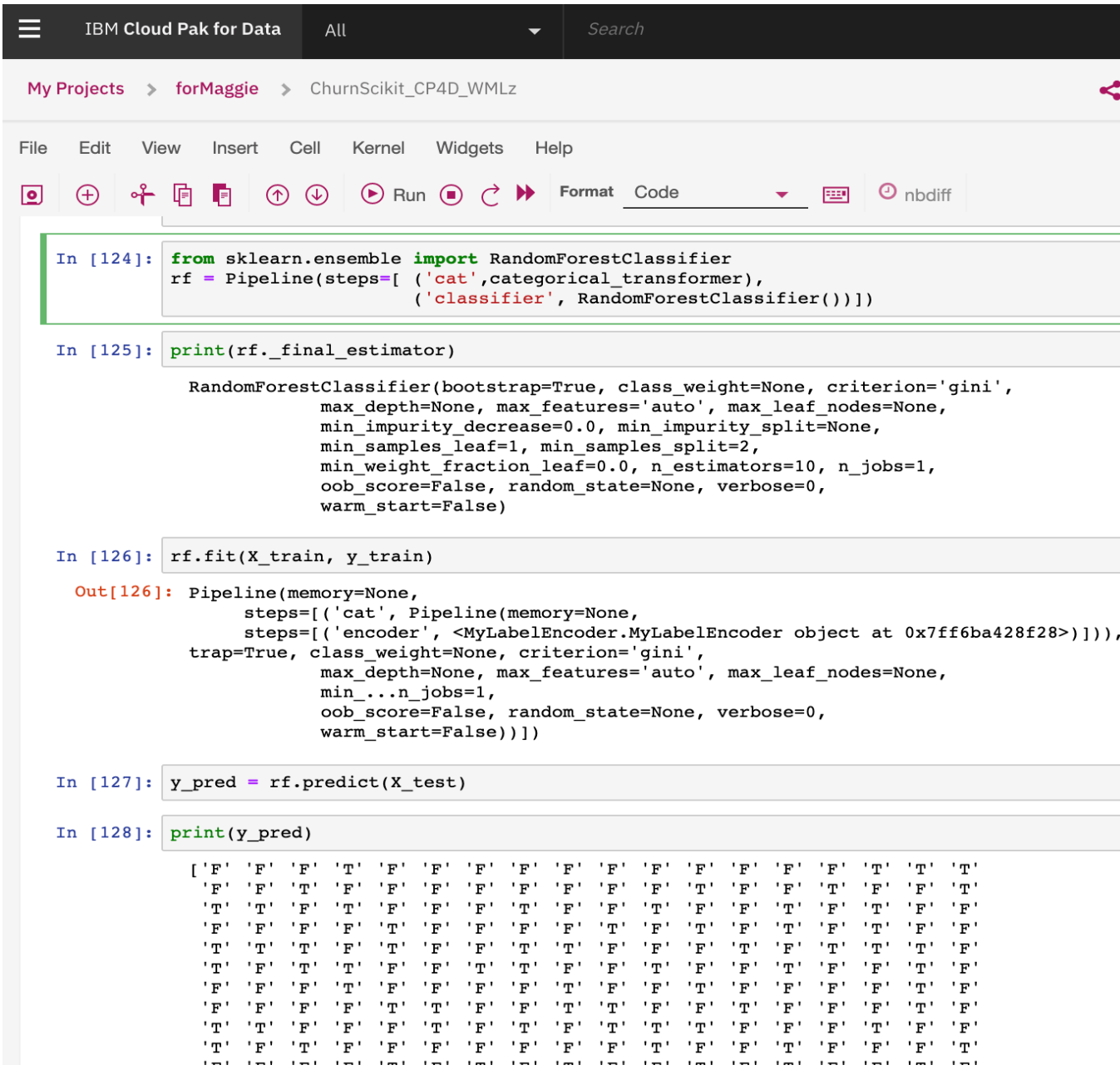
Continue on the feature engineering work for model development.



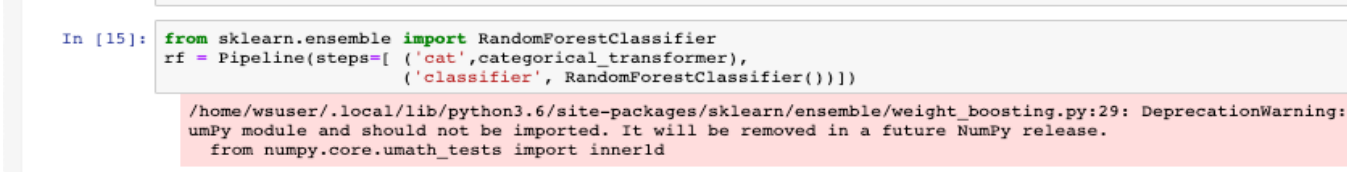
Split the data for training and testing.



__ 8. Use RandomForestClassifier as the algorithm for the pipeline training.



Note, this is a deprecation warning. Nothing to worry. Just continue.



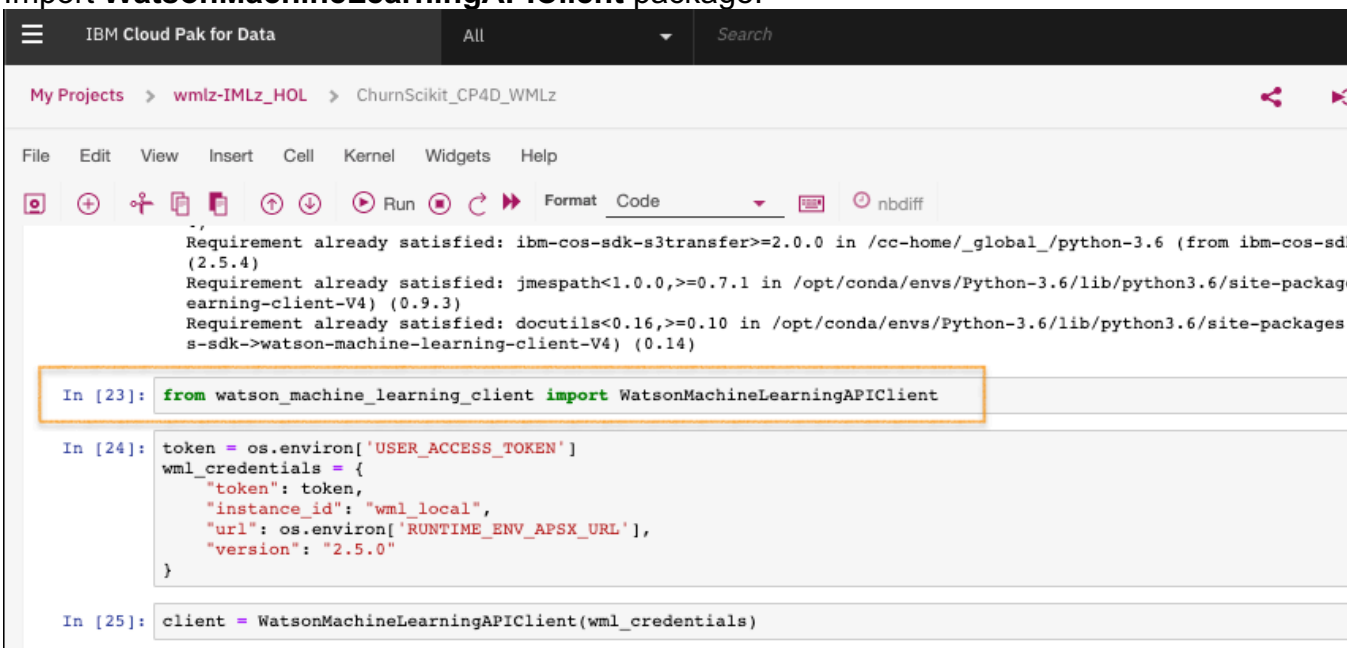
__ 9. Last part to import **WatsonMachineLearningAPIClient** package, use default space id to save the model.

Pip install **watson-machine-learning-client-V4** package if it is not there.

```
In [22]: !pip install watson-machine-learning-client-V4

         Requirement already satisfied: watson-machine-learning-client-V4 in /cc-home/_global_/python-3.6 (1.0.55)
         Requirement already satisfied: requests in /opt/conda/envs/Python-3.6/lib/python3.6/site-packages (from wats
         Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.6/lib/python3.6/site-packages (from watso
```

Import **WatsonMachineLearningAPIClient** package.



I M P O R T A N T : Update the <cp4d_login_userid> to your assigned HOL cp4d userid.

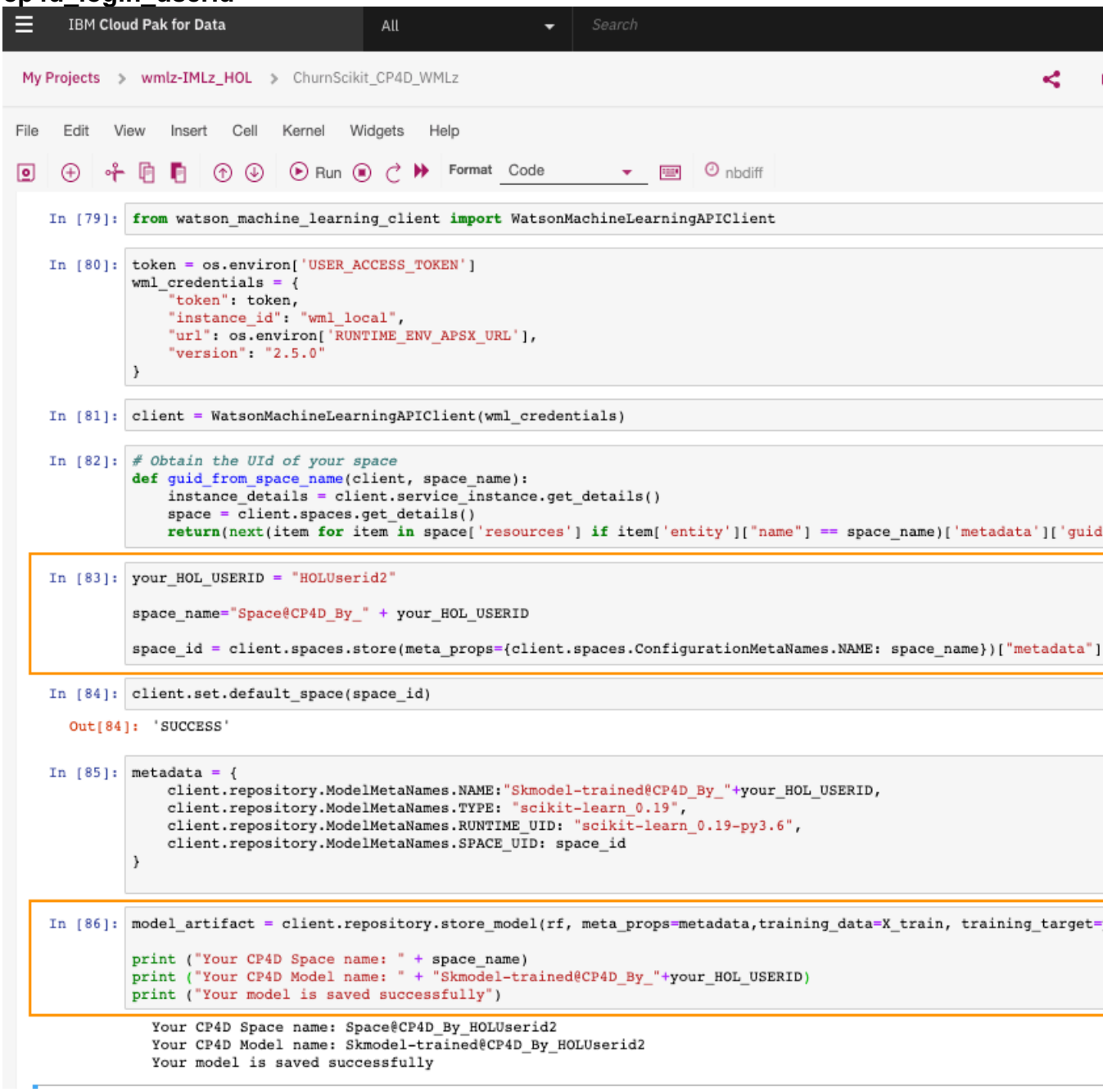
```
In [28]: your_HOL_USERID = "<cp4d_login_userid>"

         space_name="Space@CP4D_By_" + your_HOL_USERID

         space_id = client.spaces.store(meta_props={client.spaces.ConfigurationMetaNames.NAME: space_
```

Now save your trained model.
Your CP4D Space name : **Space@CP4D_By_<your_cp4d_login_userid>**
Your CP4D Model name : **Skmodel-trained@CP4D_By_<your_**

cp4d_login_userid>

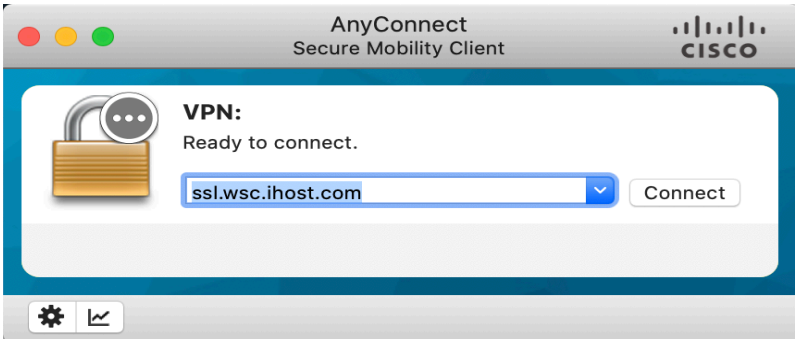


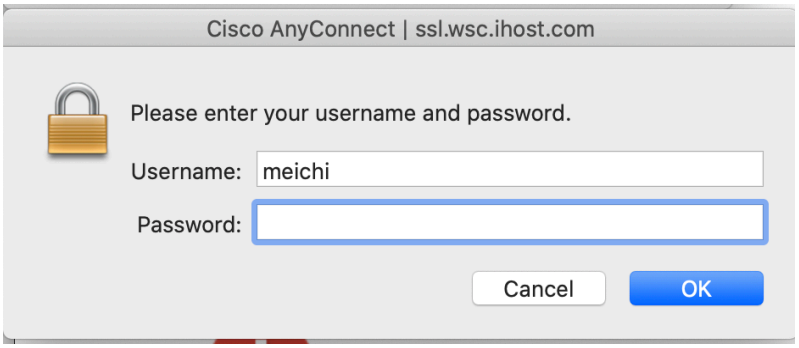
You complete the first part of the lab to get the model trained and saved successfully in CP4D.

Log in to IBM Watson Machine Learning for z/OS

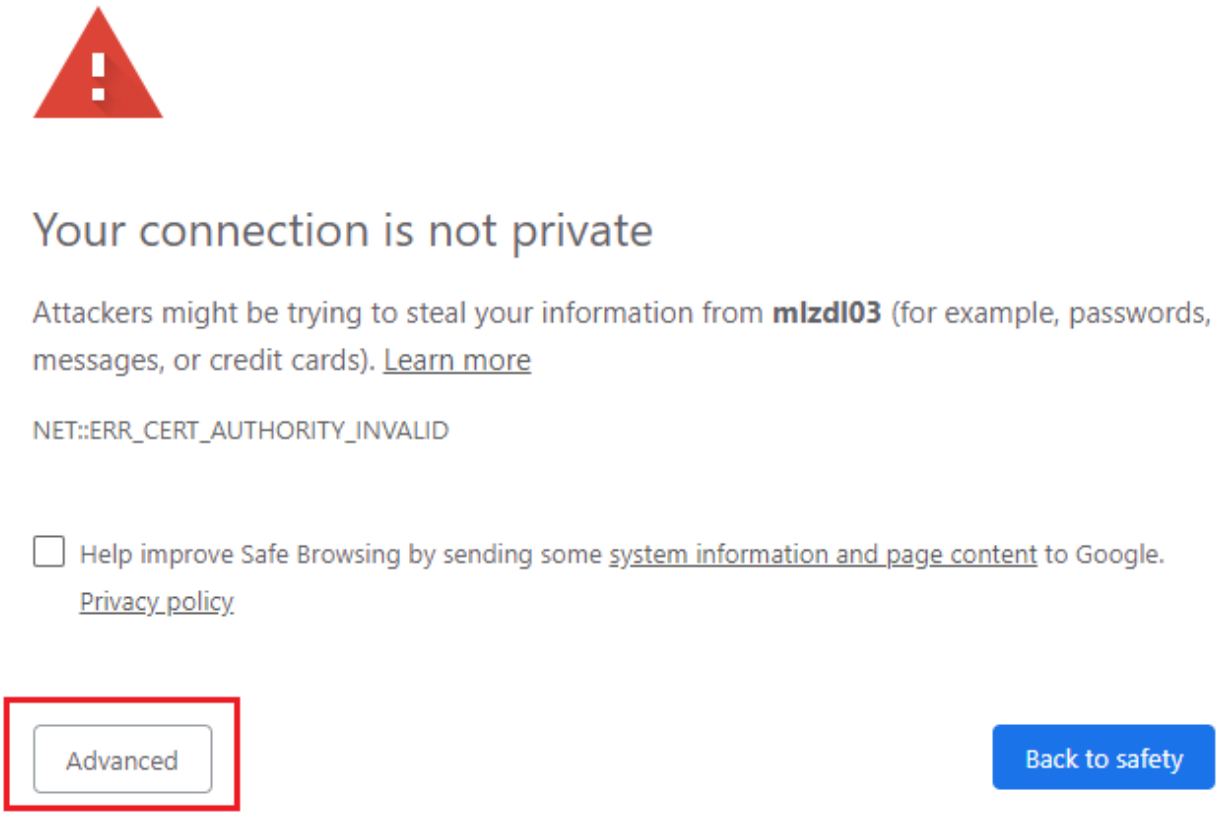
The URL and authentication credentials for the WMLz Web UI are provided on a separate worksheet. Contact the lab instructor if you did not receive a worksheet.

__ 10. Before you log in to IBM Watson Machine Learning for z/OS Web UI, you need to authenticate through **VPN** first. Open Cisco AnyConnect client and provide <wmlz_vpn_userid> & <wmlz_vpn_password> to get through VPN.

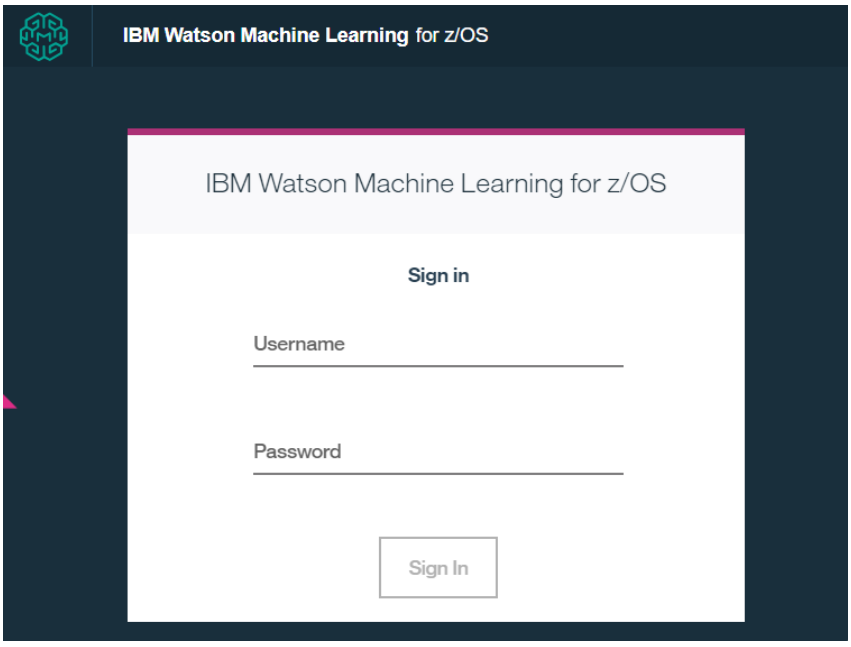




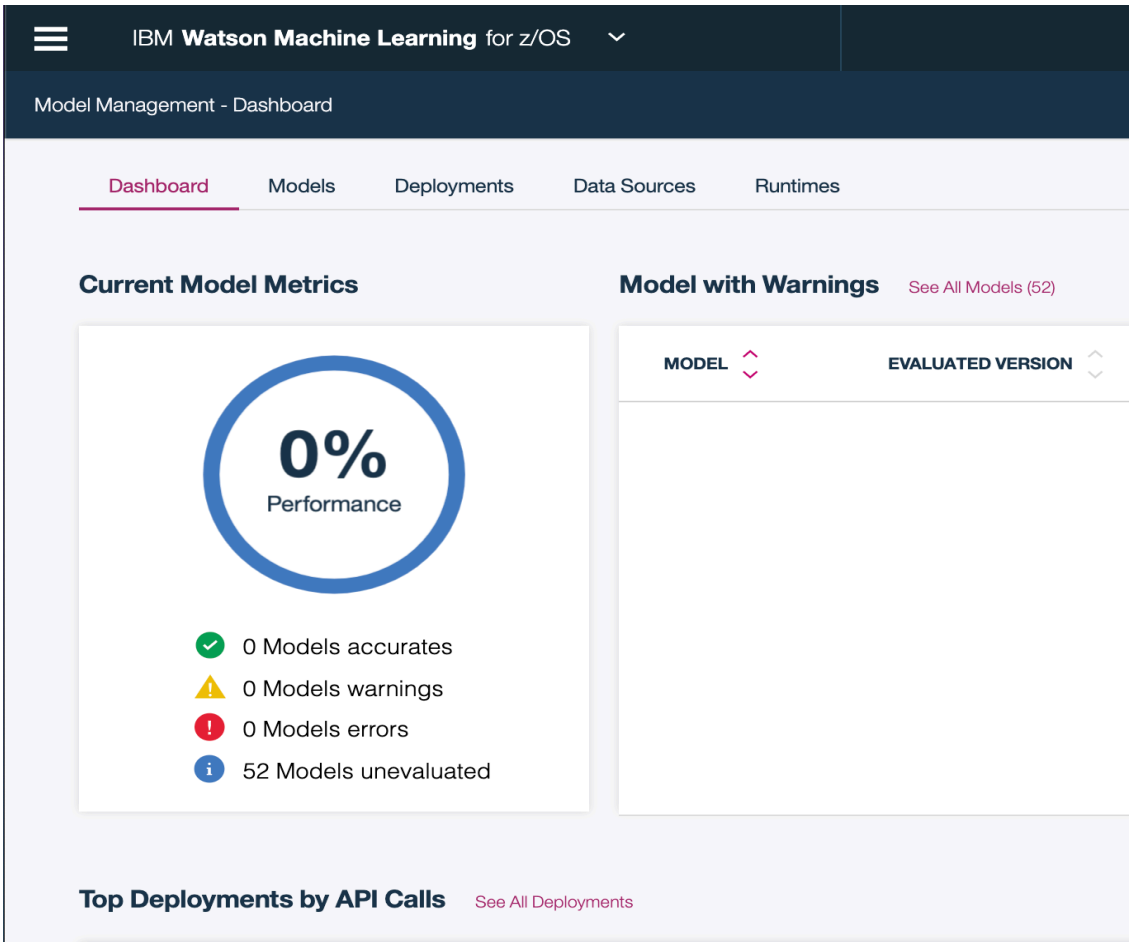
__ 11. Start the Chrome or Firefox browser, and enter the URL assigned to you, e.g. <wmlz_login_url>. For the first-time logon, you are likely to be prompted with a message regarding the security of the page you are visiting. Click “Advance” and accept the exception and proceed with the intended URL.



__ 12. Enter the <wmlz_login_userid> and <wmlz_login_password> credentials from your lab worksheet.



You should be presented with the WMLz Model Management dashboard page

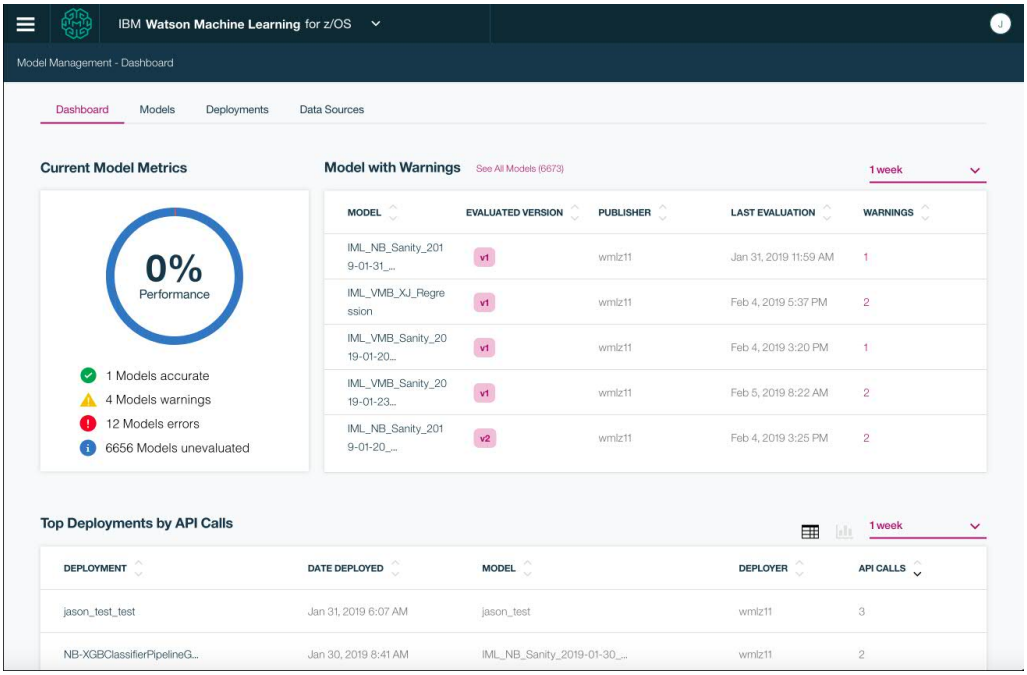


2. Model Management and Deployment in WMLz

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.

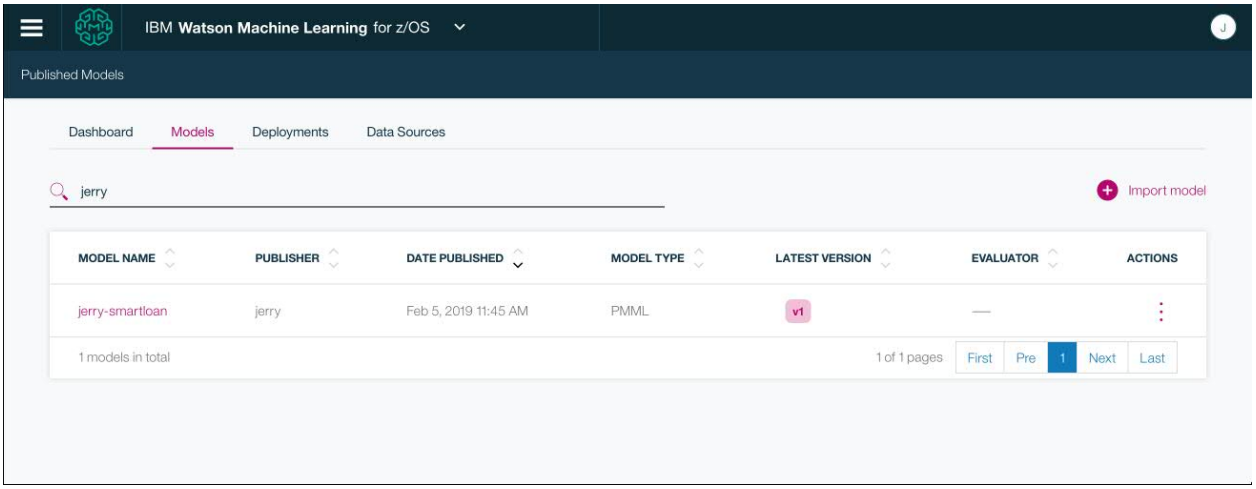
Section 1. Dashboard

__ 1. The "**Model Management Dashboard**" shows the statistical information of IBM Machine Learning platform, including the model performance and overall status.

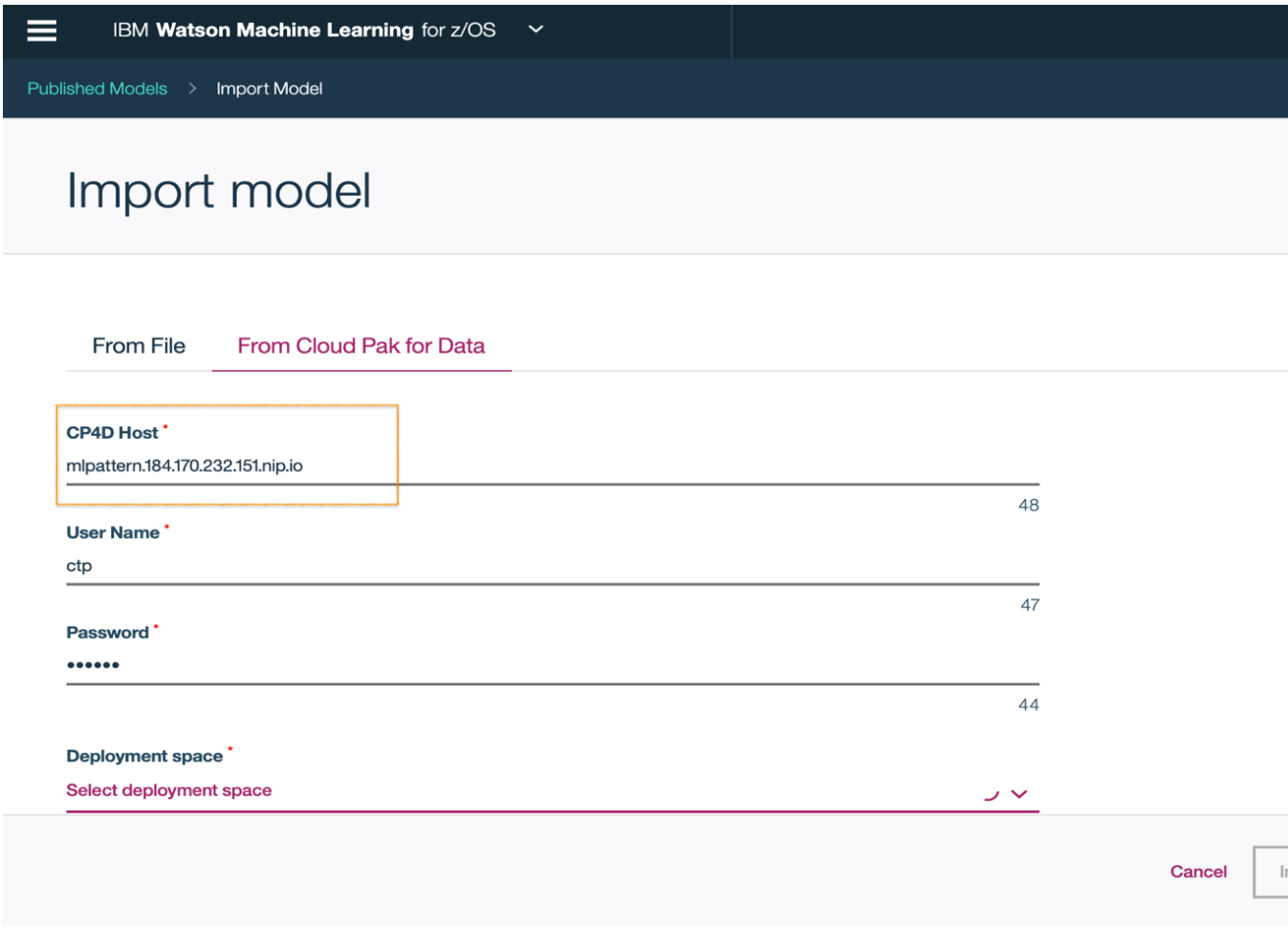


Section 2. Model Management

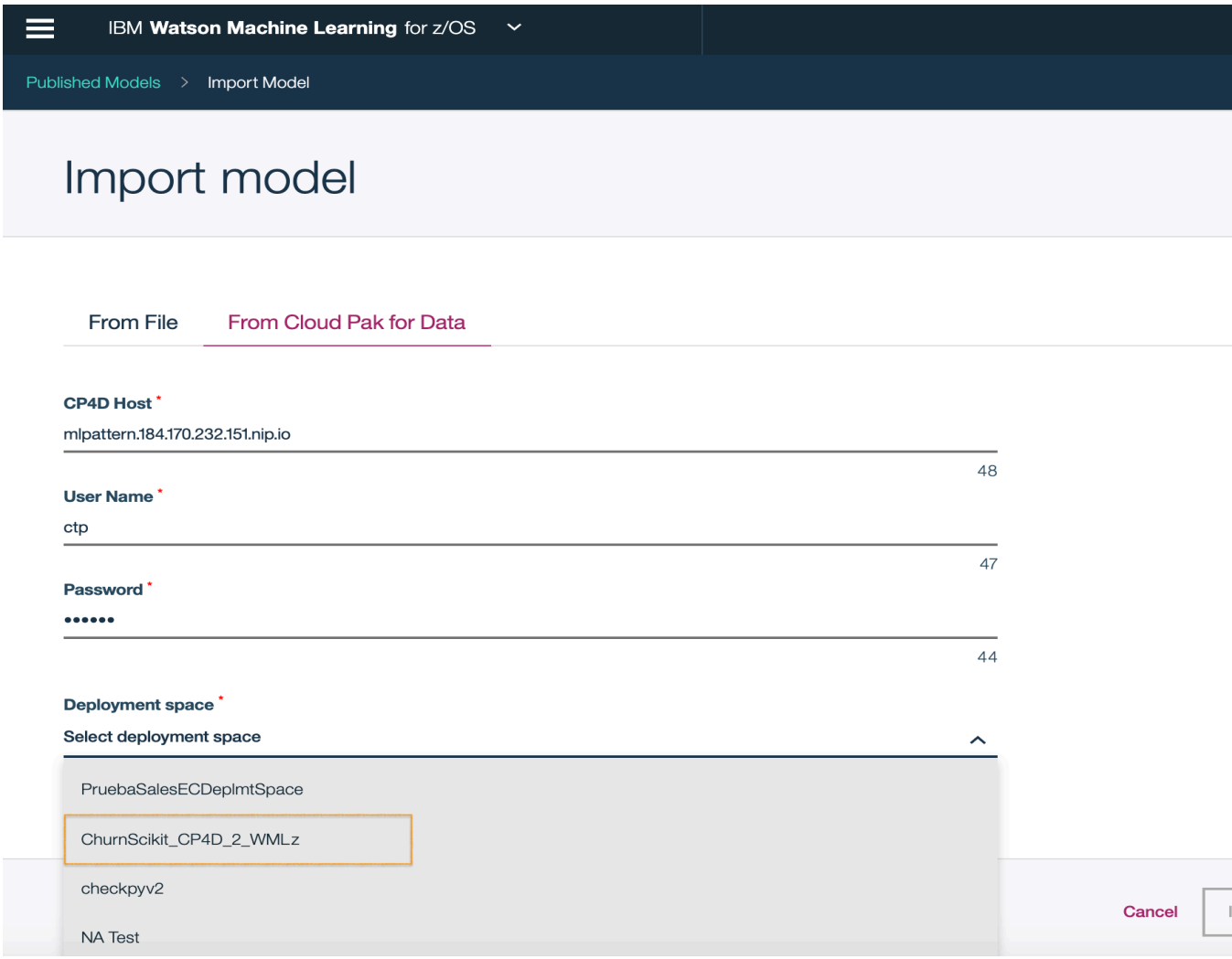
__ 2. Click the "**Models**" tab. The "**Models**" page is where you list and manage the Machine Learning models that have been created and exist in your environment. You can view the model details and model versions, retrain, delete and create deployment for a certain model. It also provides a filter to help you locate a specific model.



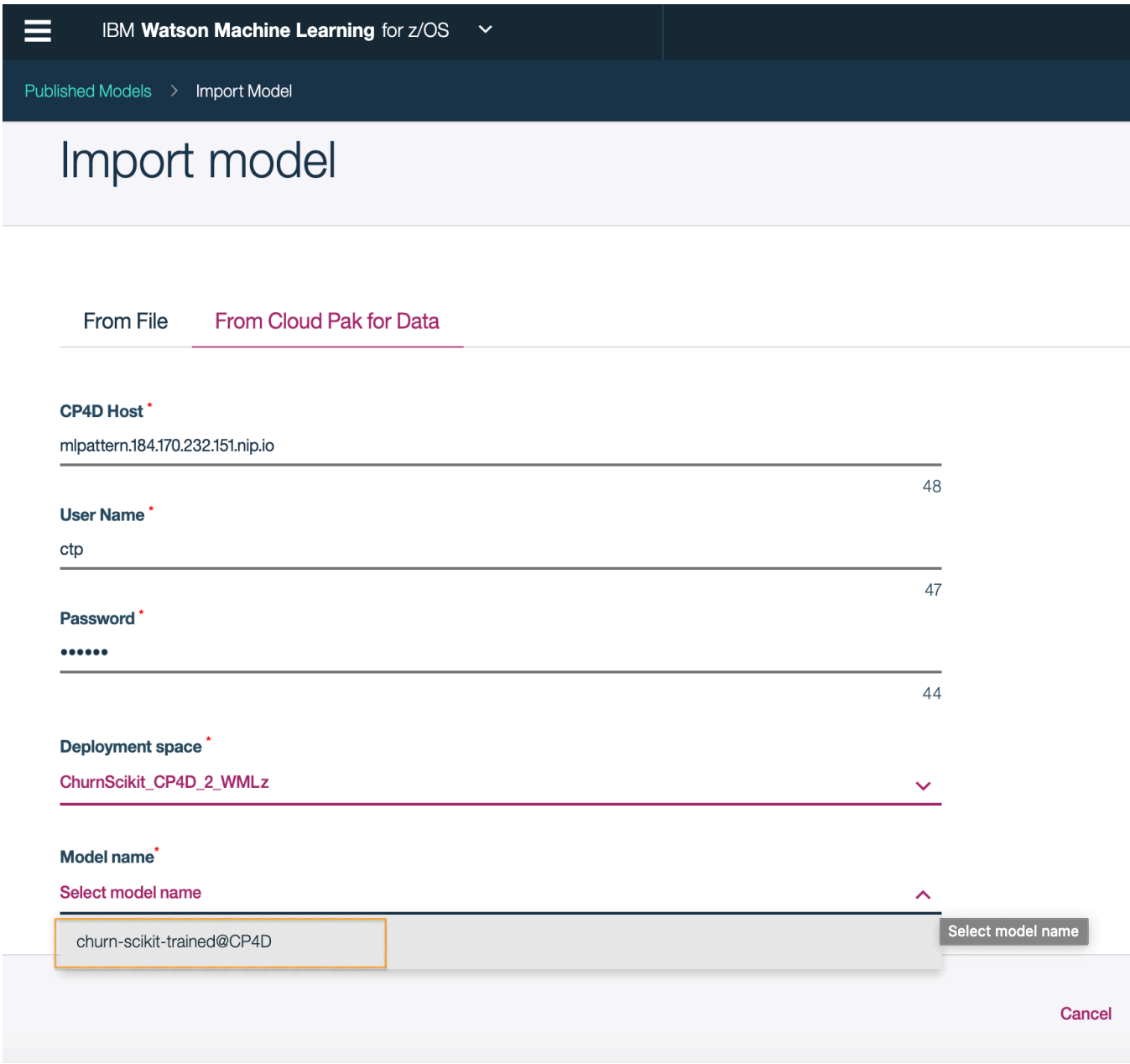
__ 3. Click on "**Import model**" to import models that are trained outside of WMLz. You see import From File or **From Cloud Pak 4 Data** option.
Choose **From Cloud Pak 4 Data** option, fill in the CP4D host, username, password.
Note, if the CP4D environment is listening on port other than 443, provide the complete CP4D host and port string in the **CP4D Host** field, such as *services-uscentral.skytap.com:13441*



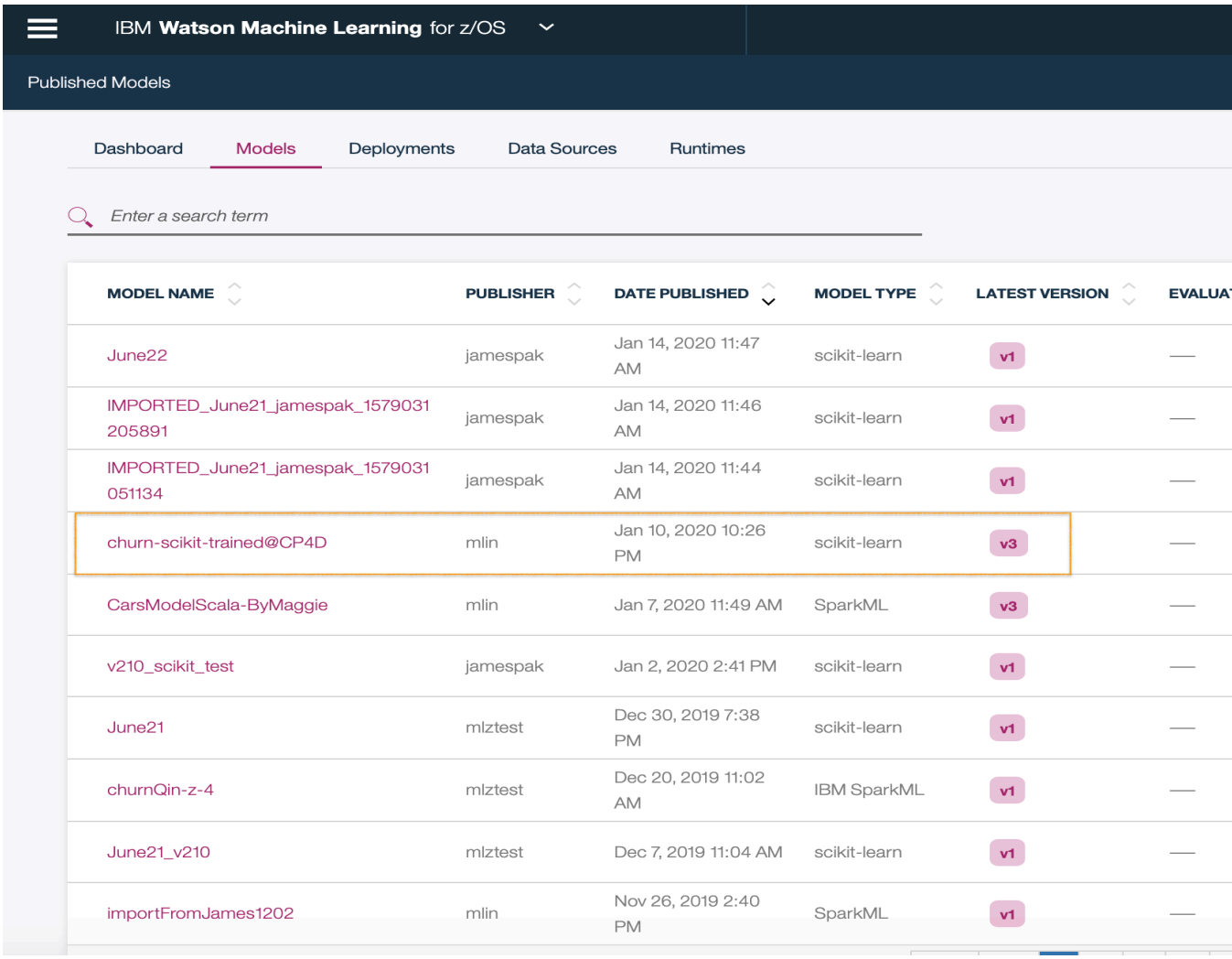
__ 4. Once it is connected to the CP4D instance, the deployment space drop-down has a list of spaces that is available for you to choose.
Choose **Space@CP4D_By_<your_cp4d_login_userid>** from the list.



__ 5. Once **Space@CP4D_By_<your_cp4d_login_userid>** deployment space is selected, go to the **Model name** drop-down list. Choose the mode **Skmodel-trained@CP4D_By_<your_ cp4d_login_userid>** model to be imported.



__ 6. 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.



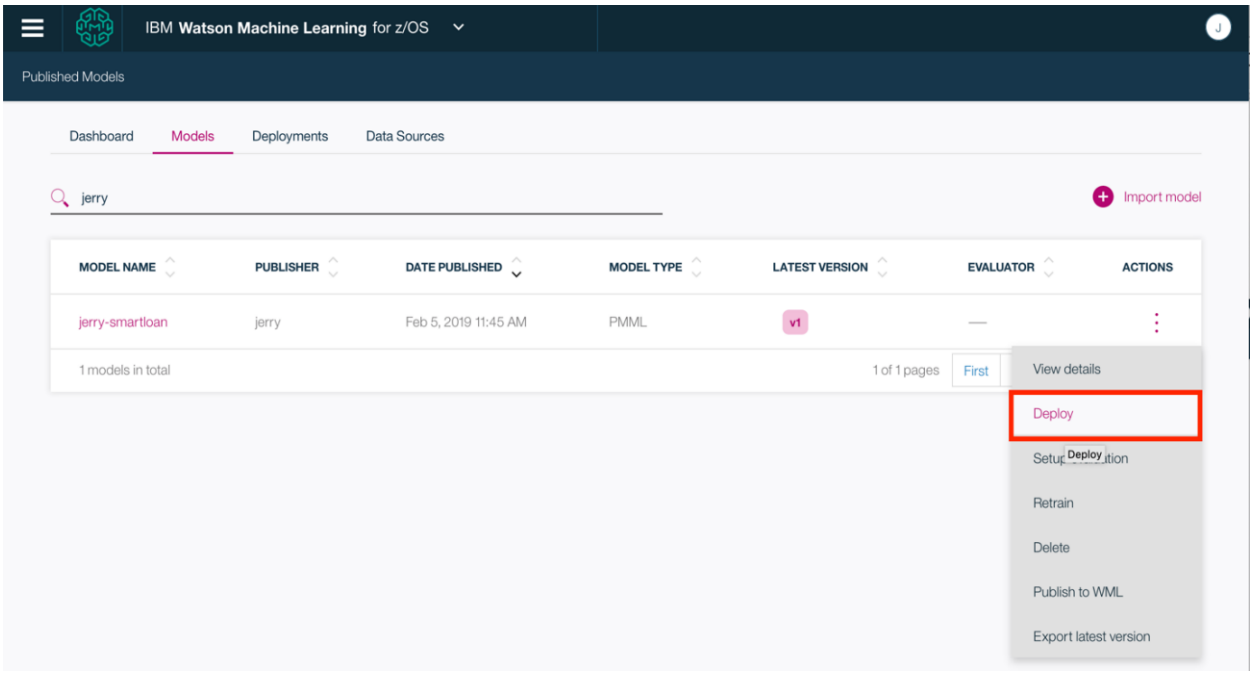
Section 3. Deployment Model to Scoring Server

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:

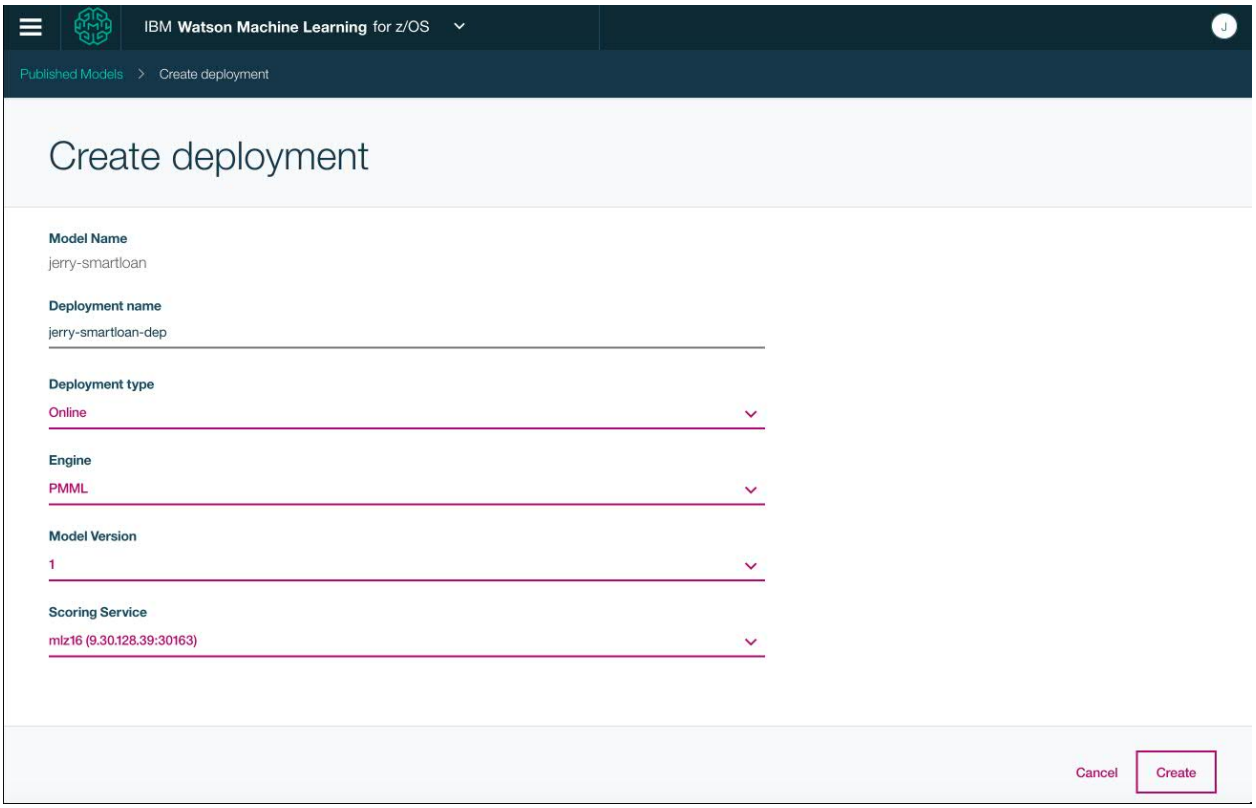
__ 7. Deploy the model

- 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 **Skmodel-trained@CP4D_By_<your_cp4d_login_userid>** is, click the actions menu and select "**Deploy**" button under "**ACTIONS.**

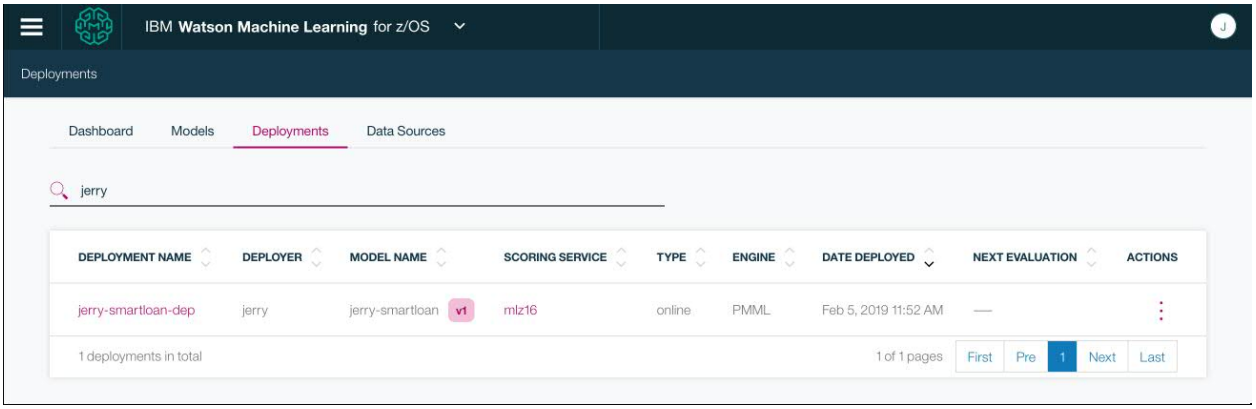
You also have the option to deploy a model created by others.



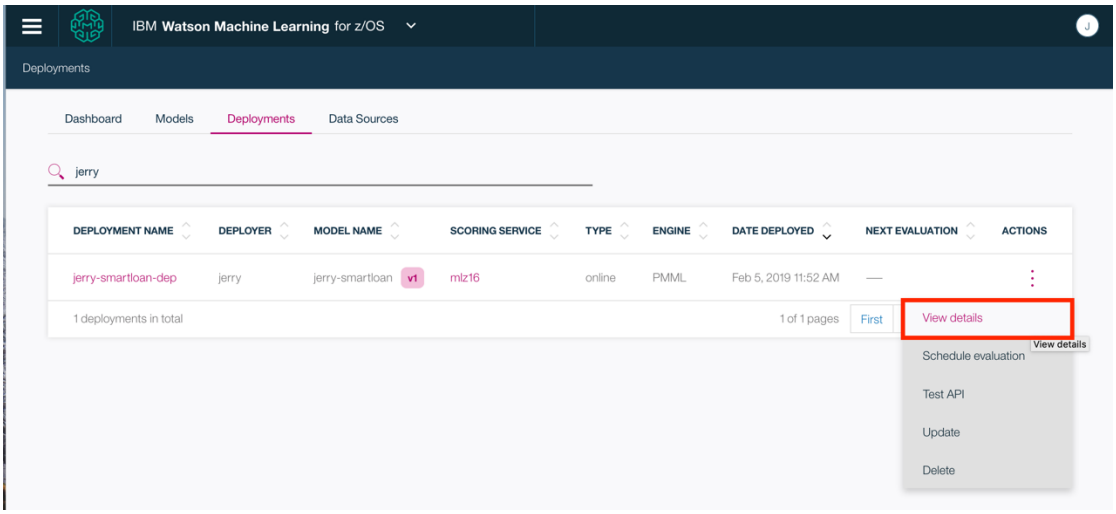
__ 8. On the "**Create deployment**" page, enter the name of the deployment and select "**Online**" for the **Type**. Choose the scoring service whose name ends in "scoring-cp4d", and then click the "**Create**" button to save.



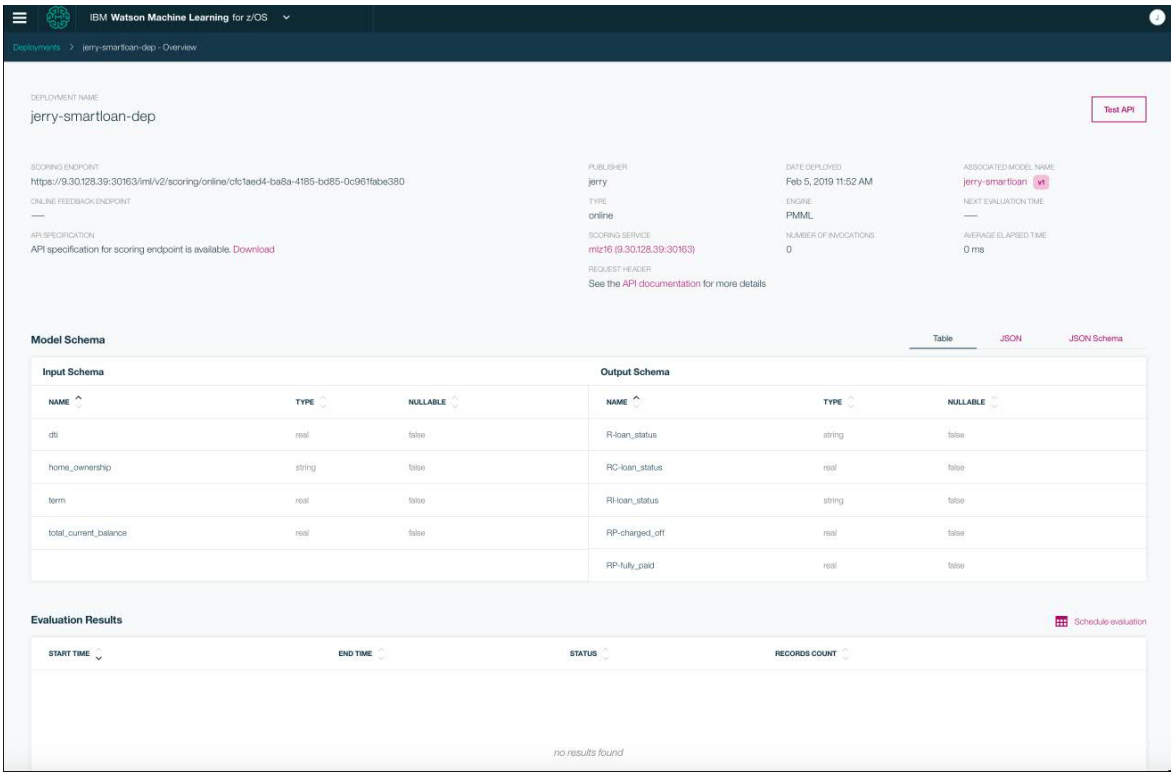
Wait until the deployment completes successfully. Then locate your deployment under the "**Deployments**" page. The "**Deployments**" page is where you view the list of models that have been deployed to your z/OS environment for scoring and prediction. You can test the model here, manage the deployment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



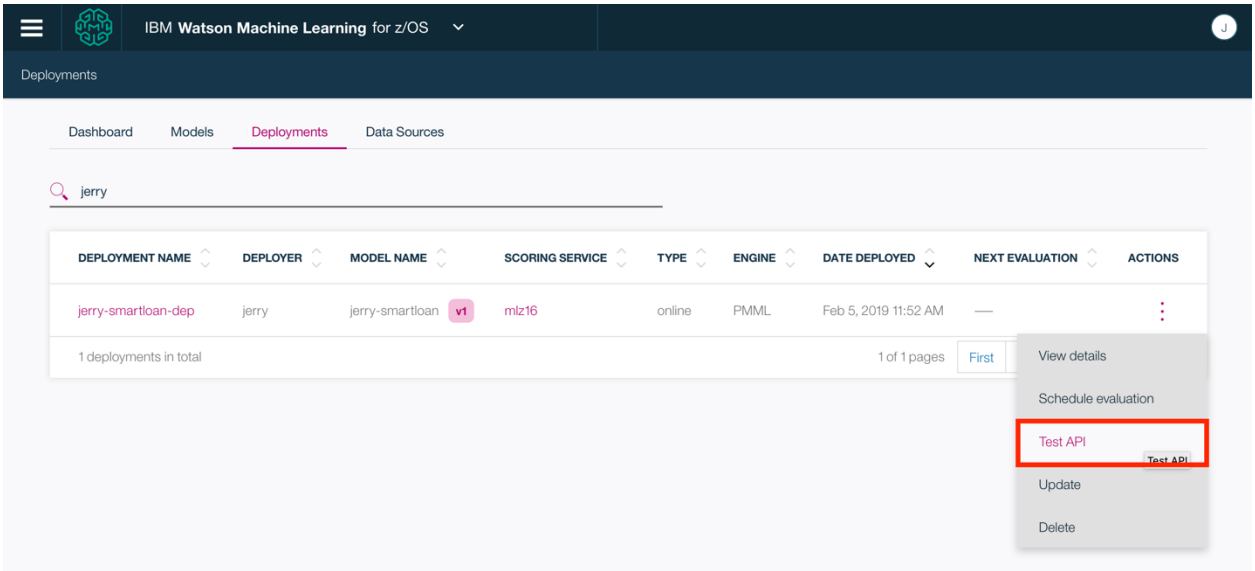
Click the "**View details**" button under "**ACTIONS**" to check the details of the deployment.



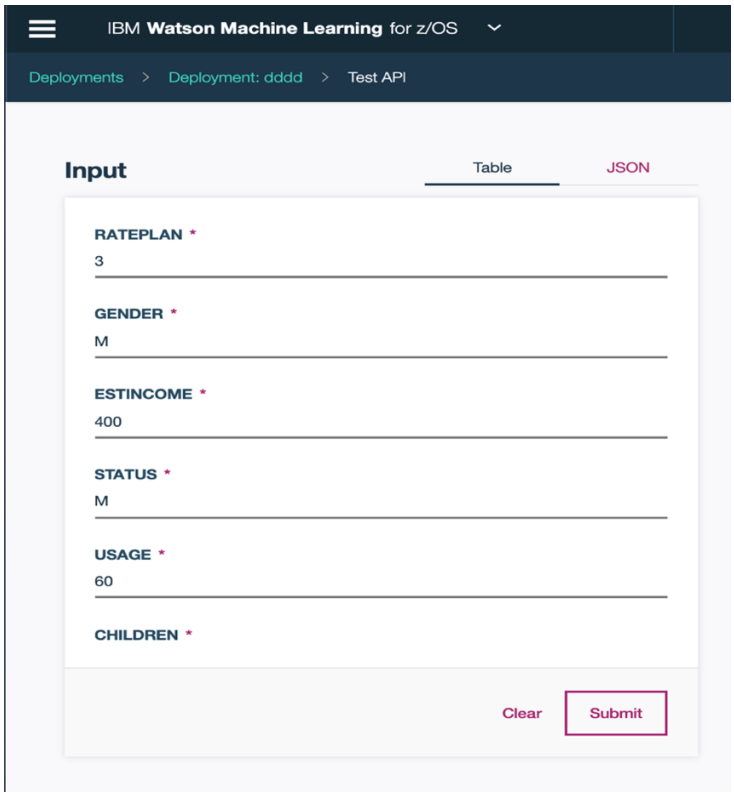
Note the API details for the RESTful API provided by the scoring service of WMLz which allows your application to call the model for online scoring. In the next step, you will test this scoring API and the model online in the Web UI. If the test is successful, you can then have your application developer 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.



__ 9. Go back to the “**Deployments**” page, click the “**Test API**” button under “**ACTIONS**”



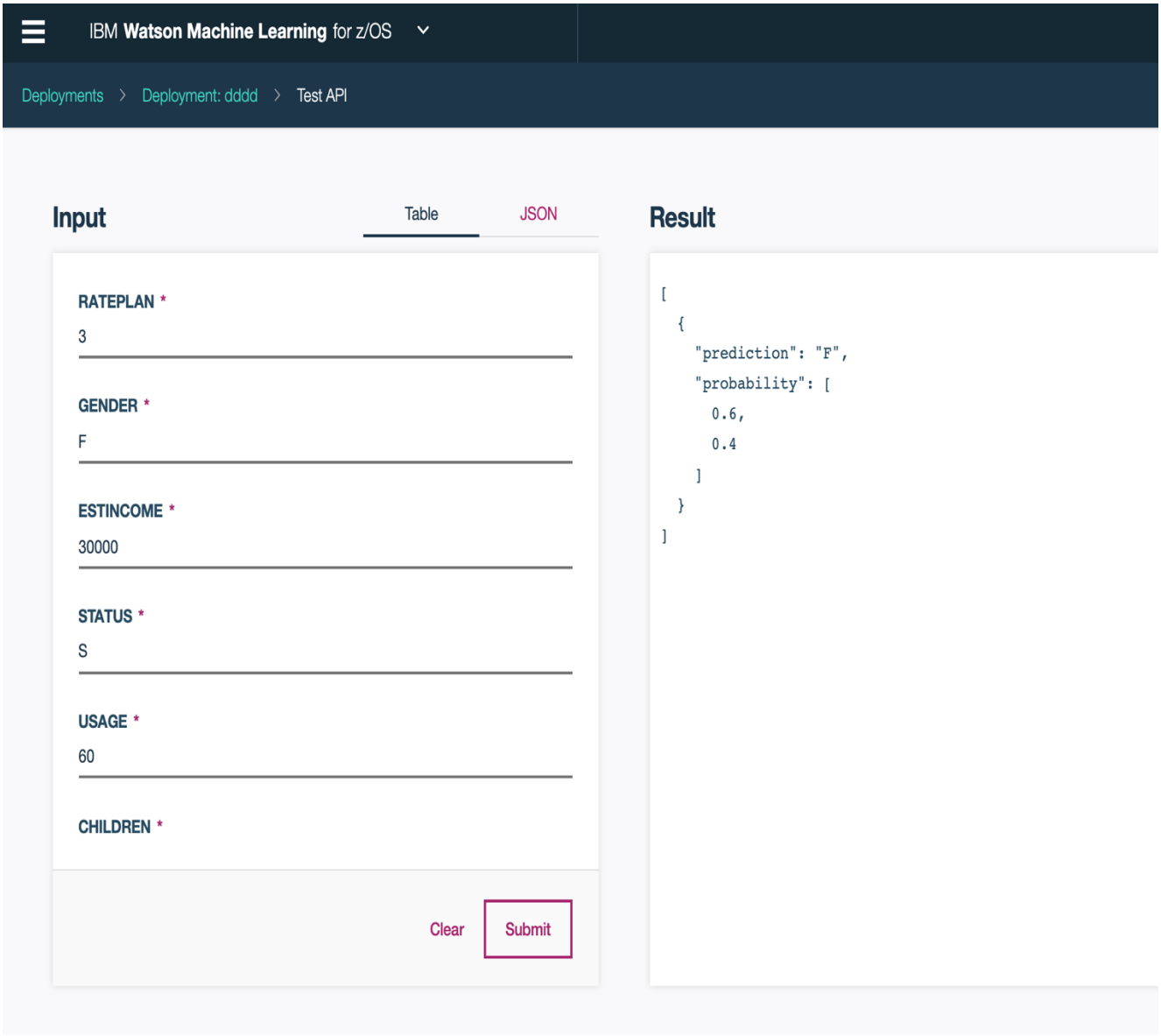
__ 10. On the “**Test API**” page, enter a record for values according to the schema of the input record of the model as below. Input values are case sensitive!



Sample Input Record:

rateplan:	3
gender:	F
estincome:	50000
status:	M
usage:	712
children:	3
age:	41
dropped:	5
localbilltype:	FreeCall
longdistance:	20
local:	55
international:	1
longdistancebilltype:	Standard
paymethod:	CC
carowner:	Y

__ 11. Click the “**Submit**” button to test the prediction. The result will be shown in “**Result**” area.



You now complete the second part of the lab to get the CP4D trained model imported to WMLz, and deployed to WMLz scoring server, and tested a scoring call against the deployed model using WMLz!

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.

