Lab-5: Heart Disease Modeling with Jupyter Notebooks

Introduction

In this lab, you use a Jupyter Notebook to train a model using the XGBoost library to classify whether a person has heart disease or not. In addition to training a model, the notebook also explains how to persist a trained model to the IBM Watson Machine Learning repository and deploy the model as a REST service.

To train and test the heart disease prediction model, you are using an open source data set published in the University of California, Irvine (UCI) Machine Learning Repository.

The notebook environment includes Python 3.7 runtime, XGBoost 0.90 and Scikit-Learn 0.23.

Objectives

Upon completing the lab, you will know how to:

1. Create a Jupyter notebook from a url.
2. Load a CSV file into Pandas DataFrame
3. Explore data using Pixiedust
4. Prepare data for training and evaluation.
5. Use Scikit Learn to create our ML pipeline. Use XGBoost to create, train, and evaluate our model.
6. Visualize the importance of features that were used to train the model.
7. Use cross validation to select optimal model hyperparameters based on a parameter grid
8. Persist the best model in the Watson Machine Learning model repository using Python client library.
9. Deploy the model for online scoring using the Python client library.
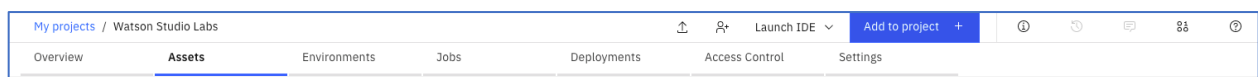
Lab Steps

Step 1 - Create a Jupyter Notebook

1. Click on the hamburger icon ☰, then click on **Projects**, and then **Watson Studio Labs** (or whatever you named the project)
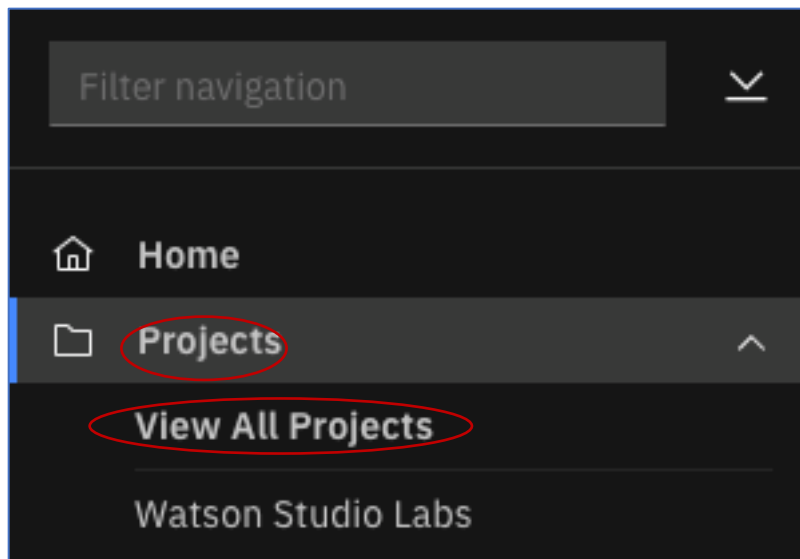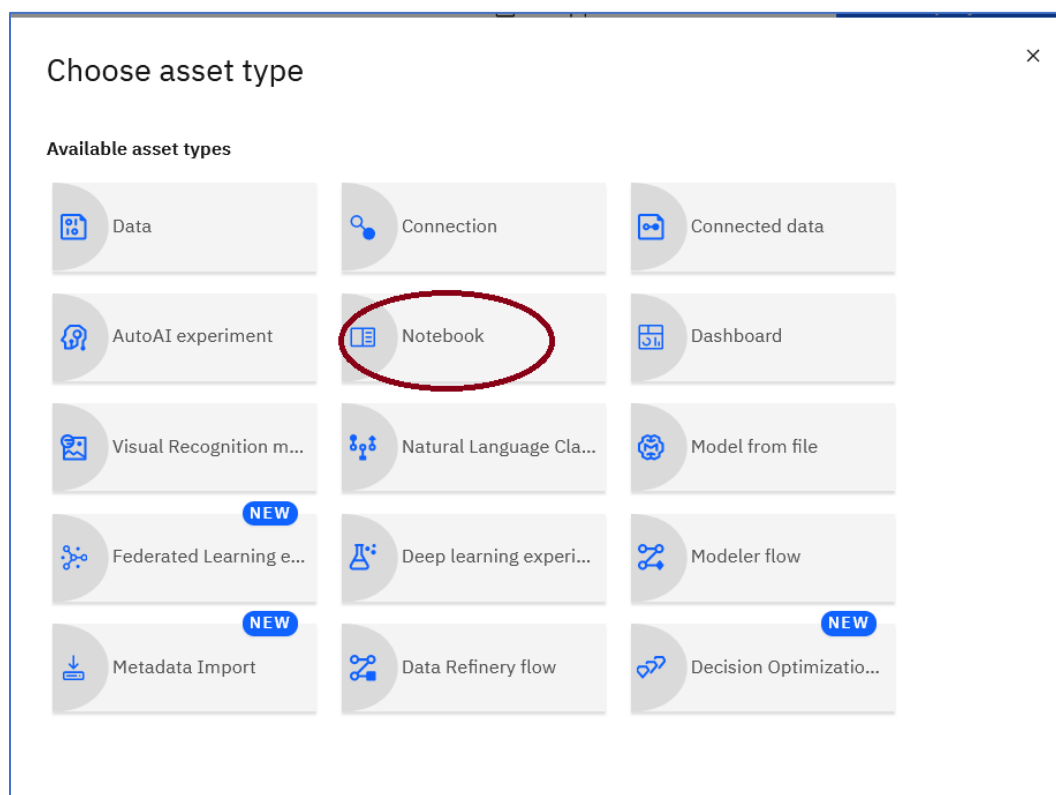
2. We are now going to create a notebook in our project. This notebook will be created from a url that points to the HeartDisease notebook in the github repository. Click the **Add to project** link.

3. Click on **Notebook**

4. Click on **From URL** under **New Notebook,** enter **Heart Disease** for the **Name**, and optionally enter a **Description**. Leave the default for the **Runtime**.

Copy hyperlink and paste the following url into the **Notebook URL** field.

https://github.com/bleonardb3/ML_POT_07-08-2021/blob/main/Lab-5/HeartDisease.ipynb

Click **Create Notebook.**

5. Before executing the code in the notebook, we need to do the following:
   a. Obtain an api key and the location of the WatsonMachineLearning service. We need to copy this information into a Notebook cell. This information is required to work with the Watson Machine Learning API. The procedure to obtain an api key and the location is described below.

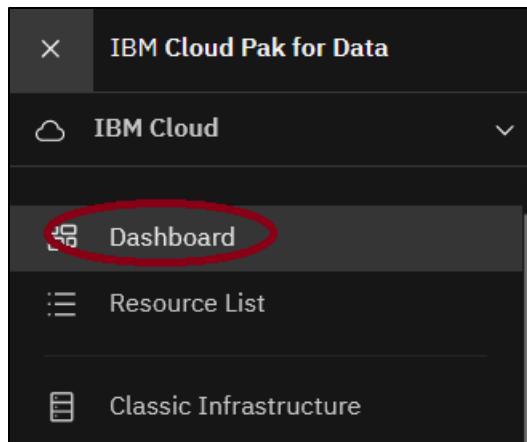6. Right-click on **IBM Cloud Pak for Data**, and the click on **Open Link in New Tab**.
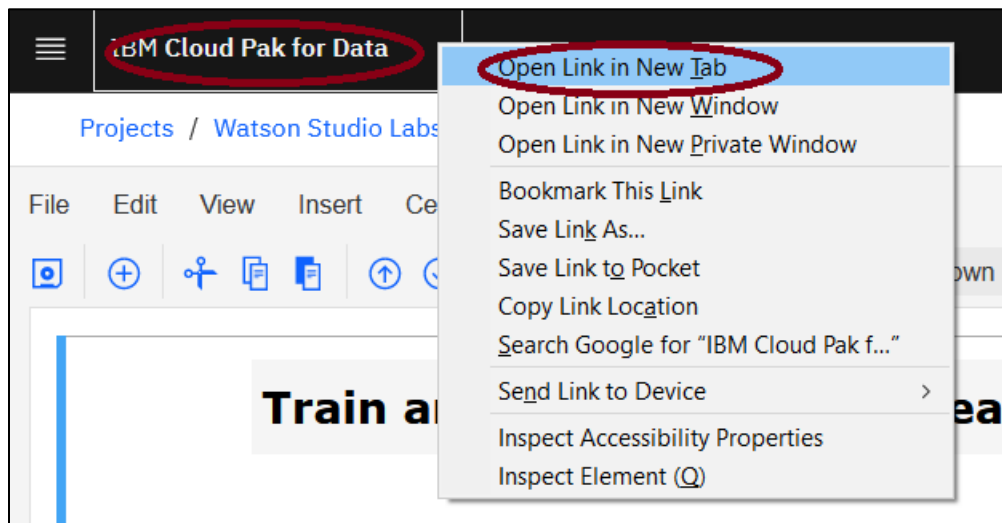
7. Click on the new **IBM Cloud Pak for Data** browser tab.

8. Click on the hamburger icon ☰, and then scroll down to the bottom to click on **IBM Cloud**.
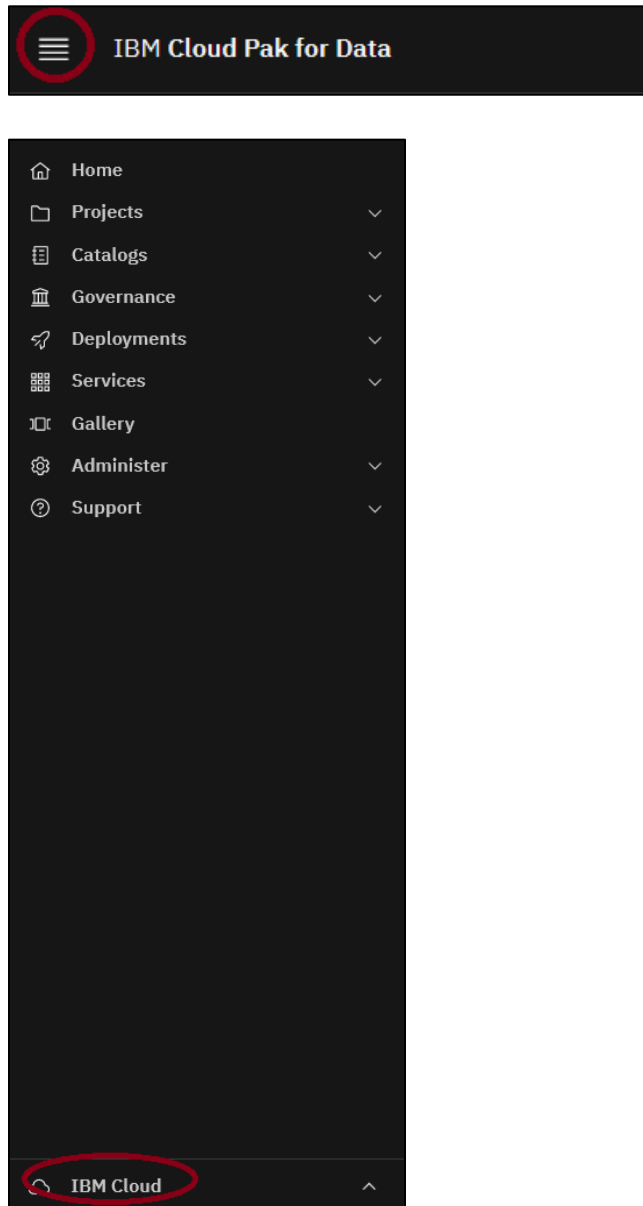
9. Click on **Dashboard**

10. Login again if prompted. Click on the IBM Command Shell icon .

11. Wait for the shell to be initialized. To create an api key, copy and paste the following line next to the command shell prompt and press the <Enter> key.

```
ibmcloud iam api-key-create API_KEY_NAME
```

```
You can view this entry in the screen image below highlighted in
maroon.
```

```
The newly created api key is shown highlighted in blue below. This key
is unique to my account. Your key will be unique to your account.
```

```
To get the location of the WatsonMachineLearning service instance, copy
and paste the following line next to the command shell prompt and press
the <Enter> key.
```

```
ibmcloud resource service-instance WatsonMachineLearning
```

```
You can view this entry in the screen image below also highlighted in
maroon.
```

```
The location of the WatsonMachineLearning instance is shown highlighted
in blue below. My instance location is shown to be in us-south.
```

```
Welcome to IBM Cloud Shell!
Image version: 1.0.9
Help us improve future releases by clicking Feedback to share your experience!

Note: Your Cloud Shell session is running in Dallas (us-south). Your workspace includes 500 MB of temporary storage. This session will close after
, your workspace data is removed. To track your usage, go to Usage quota in the Cloud Shell menu.

Tip: Enter 'ibmcloud' to use the IBM Cloud CLI. The Dallas (us-south) region is targeted by default. You can switch the region by running 'ibmclou

wsuser56000@cloudshell:~$ ibmcloud iam api-key-create API_KEY_NAME
Creating API key API_KEY_NAME under 1aab03f27a8b47e2be36e68574e8fcce as wsuser56000@gmail.com...
OK
API key API_KEY_NAME was created

Please preserve the API key! It cannot be retrieved after it's created.

ID              ApiKey-413e0d2a-cadc-4388-b4e7-2da4d02b8de9
Name            API_KEY_NAME
Description
Created At      2020-10-28T03:24+0000
API Key         WeqpHa-Av0wn2PhEE_CXGcGK1lqkcJevwPQI51mfT19t
Locked          False
wsuser56000@cloudshell:~$ ibmcloud resource service-instance WatsonMachineLearning
Retrieving service instance WatsonMachineLearning in all resource groups under account Steven Doe's Account as wsuser56000@gmail.com...
OK

Name:           WatsonMachineLearning
ID:             crn:v1:bluemix:public:pm-20:us-south:a/1aab03f27a8b47e2be36e68574e8fcce:93d46d98-4f8a-4547-8838-cbf1f9065455::
GUID:           93d46d98-4f8a-4547-8838-cbf1f9065455
Location:       us-south
Service Name:   pm-20
Service Plan Name:  lite
Resource Group Name: Default
State:          active
Type:           service instance
```

12. Copy and paste your api key and location values into the notebook cell below **Setup**. The screen below shows the api key and location for my setup. Click on the Heart Disease browser tab to go back to the notebook.

1. Setup

In order to interface to the Watson Machine Learning service we need to obtain an api key and find the location of the service instance. The procedure is written in the lab instructions.

```
api_key = "WeqpHa-Av0wn2PhEE_CXGcGK1lqkcJevwPQI51mfT19t"
location = "us-south"
```

Before you execute the sample code in this notebook, you must download the **Heart Disease Data Set** data in the Notebook's local filesystem

Download Heart Disease Data Set to Notebook's local filesystem

13. Return to the top of the notebook, read through the documentation in the beginning and then select the first code cell to execute.

1. Setup

In order to interface to the Watson Machine Learning service we need to obtain an api key and find the location of the service instance. The procedure is written in the lab instructions.

```
api_key = "WeqpHa-Av0wn2PhEE_CXGcGK1lqkcJevwPQI51mfT19t"
location = "us-south"
```

Before you execute the sample code in this notebook, you must download the **Heart Disease Data Set** data in the Notebook's local filesystem

Download Heart Disease Data Set to Notebook's local filesystem

For those not familiar with Jupyter notebooks, read below.

A Jupyter notebook consists of a series of cells. These cells are of 2 types (1) documentation cells containing markdown, and (2) code cells (denoted by a bracket on the left of the cell) where you write Python code, R, or Scala code depending on the type of notebook. Code cells can be run by putting the cursor in the code cell and pressing **<Shift><Enter>** on the keyboard. Alternatively, you can execute the cells by clicking on **Run icon** on the menu bar that will run the current cell (where the cursor is located) and then select the cell below. In this way, repeatedly clicking on **Run** executes all the cells in the notebook. When a code cell is executed the brackets on the left change to an asterisk '*' to indicate the code cell is executing. When completed, a sequence number appears. The output, if any, is displayed below the code cell.

14. Execute each of the notebook cells in order (either by typing in **<Shift><Enter>** or using the **Run** menu option). Read the notebook documentation to gain an understanding of the code that is executing. **When all the cells in the notebook have been successfully executed, please return to this document, and continue with the below steps.**

    Note: The cells shown in the following image will require you to copy and paste the Watson Studio Labs **space id** into the cell that follows it.

    Before:

    After:

15. The notebook built, trained, saved, and deployed a machine learning model. Click Watson Studio Labs to exit out of the notebook.
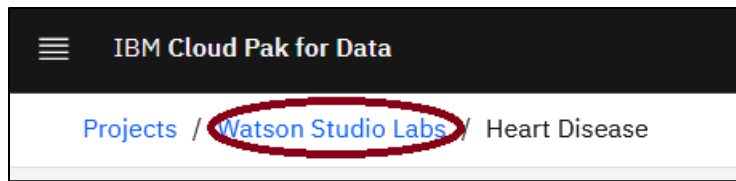
16. Click on the hamburger icon ☰, then click on **Deployments**, and then **Watson Studio Labs**.
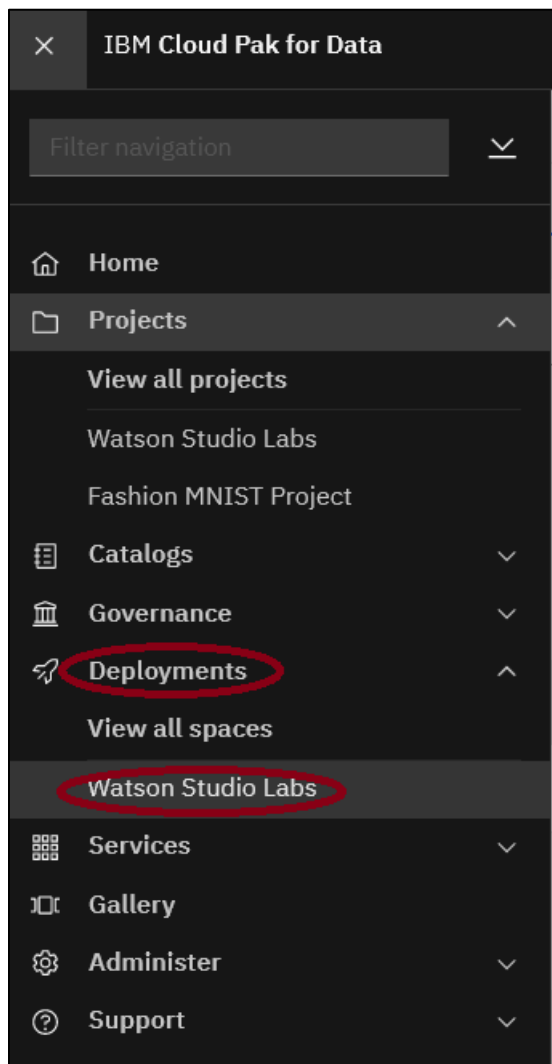
17. You can see the Heart Disease model asset that was saved programmatically to the deployment space model repository in the notebook.

Models (1)						Import model +
Name	**Type**	**Software specification**		**Tags**	**Last modified**	↓
◈ Heart Disease	scikit-learn_0.23	default_py3.7			Mar 20, 2021 3:20 PM	

18. Click on **Deployments**

Watson Studio Labs

Assets   Deployments   Jobs   Manage

🔍 What assets are you looking for?

∨   Models (1)   Import model +

Name	**Type**	**Software specification**	**Tags**	**Last modified**	↓
◈ Heart Disease	scikit-learn_0.23	default_py3.7		Mar 20, 2021 3:20 PM	

19. The Heart Disease Deployment listed was generated programmatically from the notebook using the Watson Machine Learning APIs. It is an online deployment. Click on **Heart Disease Deployment.**

Watson Studio Labs

Assets   **Deployments**   Jobs   Manage

🔍 What deployments are you looking for?

Deployments (1)

Name	**Type**	**Status**	**Asset**	**Last modified**	↓
Heart Disease Deployment	Online	✅ Deployed	Heart Disease	Mar 20, 2021 3:23 PM	

20. The **API Reference** panel provides sample code in various programming languages and the scoring endpoint to be used when invoking the deployed model via a RESTful interface. Click on **Test**.

21. The **Test** panel enables testing the deployed model. Copy and then paste the following json between the [] after input_data:

```
{"fields":
["age","sex","cp","restbp","chol","fbs","restecg","thalach","exa
ng","oldpeak","slope","ca","thal"], "values":
[[52.0,1.0,1.0,118.0,186.0,0.0,2.0,190.0,0.0,0.0,2.0,0.0,6.0]]}
```

22. Click on **Predict**. The model predicts that this patient does not have heart disease with confidence of 98%.

You have completed the Lab!

- ✓ Loaded a CSV file into Pandas DataFrame.
- ✓ Explored data using Pixiedust
- ✓ Prepared data for training and evaluation.
- ✓ Created, trained, and evaluated a XGBoost model.
- ✓ Visualized the importance of features that were used to train the model.
- ✓ Used cross validation to select optimal model hyperparameters based on a parameter grid
- ✓ Persisted the best model in Watson Machine Learning repository using the Python client library.
- ✓ Deployed the model for online scoring using the Python client library
- ✓ Visualized Deployment in the Watson Studio UI.
- ✓ Tested the deployment