Watson Studio: 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 3.5 runtime, XGBoost 0.6 and Scikit-Learn 0.17.

Objectives

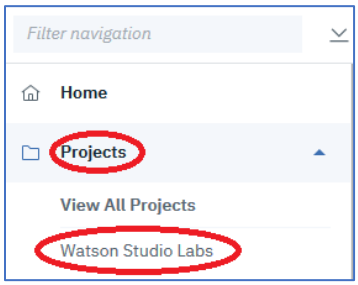
Upon completing the lab, you will know how to:

1. Load a CSV file into Pandas DataFrame.
2. Data exploration using Pixiedust
3. Prepare data for training and evaluation.
4. Create, train, and evaluate a XGBoost model.
5. Visualize the importance of features that were used to train the model.
6. Use cross validation to select optimal model hyperparameters based on a parameter grid
7. Persist best model in Watson Machine Learning repository using Python client library.
8. Deploy the model for online scoring using the Watson Machine Learning's REST APIs

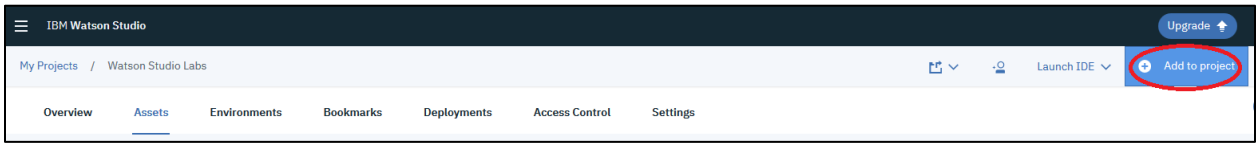
Lab Steps

Step 1 - Create a Jupyter Notebook

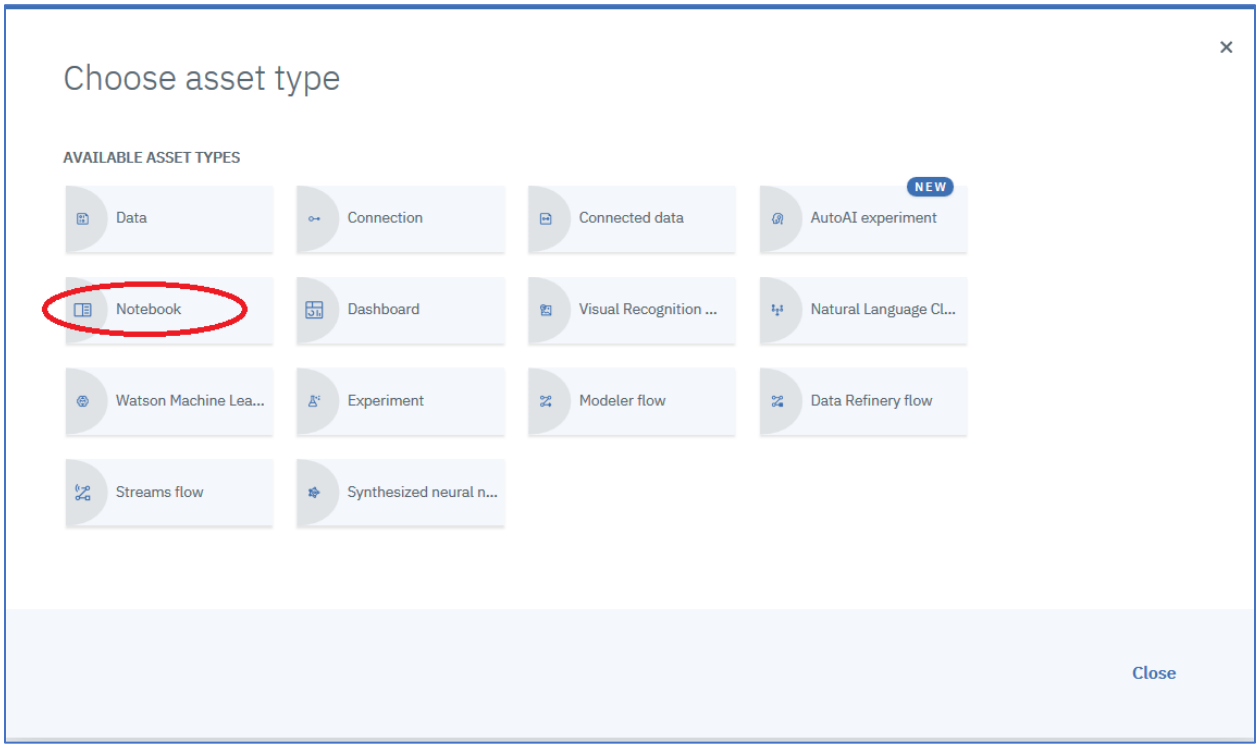
1. Click on the hamburger icon, then click on **Projects**, and then **Watson Studio Labs** (or whatever you named the project)



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**



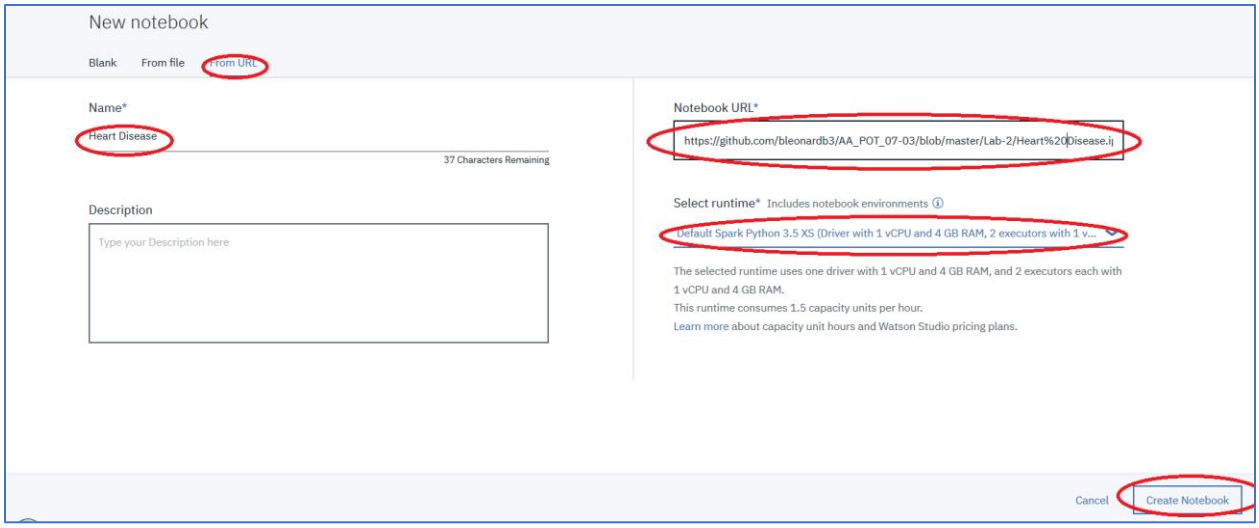
4. Click on **From URL** under **New Notebook,** enter **Heart Disease** for the **Name**, optionally enter a **Description**, cut and paste the following url into the **Notebook URL** field.

   https://github.com/bleonardb3/AA_POT_07-02/blob/master/Lab-2/Heart%20Disease.ipynb

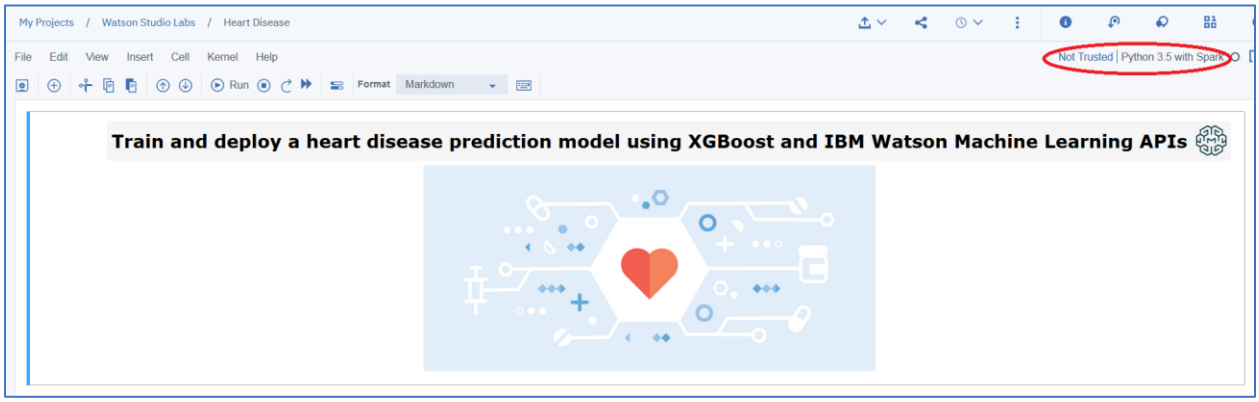
   Select the Runtime.

   MAKE SURE TO SELECT Default Spark Python 3.5 XS (Driver with 1vCPU ...

   Click **Create Notebook.**



5. Please make sure the notebook has **Python 3.5 with Spark** in the top right corner.



6. 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.

7. Execute each of the notebook code cells in order. Read the notebook documentation to gain an understanding of the code that is executing.

You have completed Lab-2!

✓ 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 Python client library.
- ✓ Deployed the model for online scoring using the Watson Machine Learning's REST APIs