Lab-3: Explainability using AIX360

Introduction

Black box machine learning models that cannot be understood by people, such as deep neural networks and large ensembles, are achieving impressive accuracy on various tasks. However, as machine learning is increasingly used to inform high stakes decisions, explainability and interpretability of the models are becoming essential.

AI Explainability 360 is an open source toolkit developed by IBM Research, that can help explain why a machine learning model came to a decision. This toolkit includes algorithms that span the different dimensions of ways of explaining along with proxy explainability metrics.

For more information see links below:

```
AIX360 Demo: https://aix360.mybluemix.net
AIX360 GitHub: https://github.com/IBM/AIX360/
AIX360 API Docs: https://aix360.readthedocs.io/en/latest/
```

Objectives

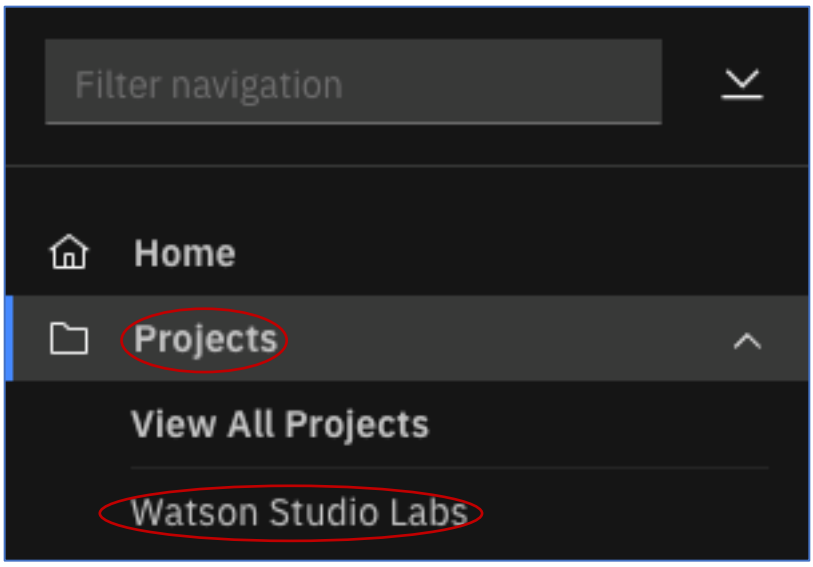
Upon completing the lab, you will learn how to:

1. Load a dataset using a download link
2. Create, train, and evaluate a XGBoost model
3. Use Protodash Algorithm to extract similar examples and compare with the current patient's case

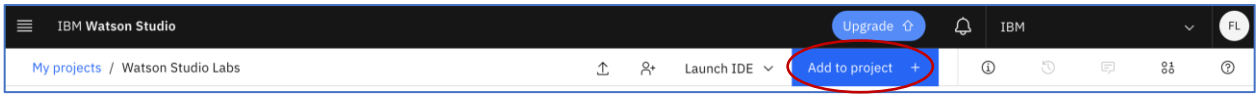
Lab Steps

Step 1 - Create a Jupyter Notebook

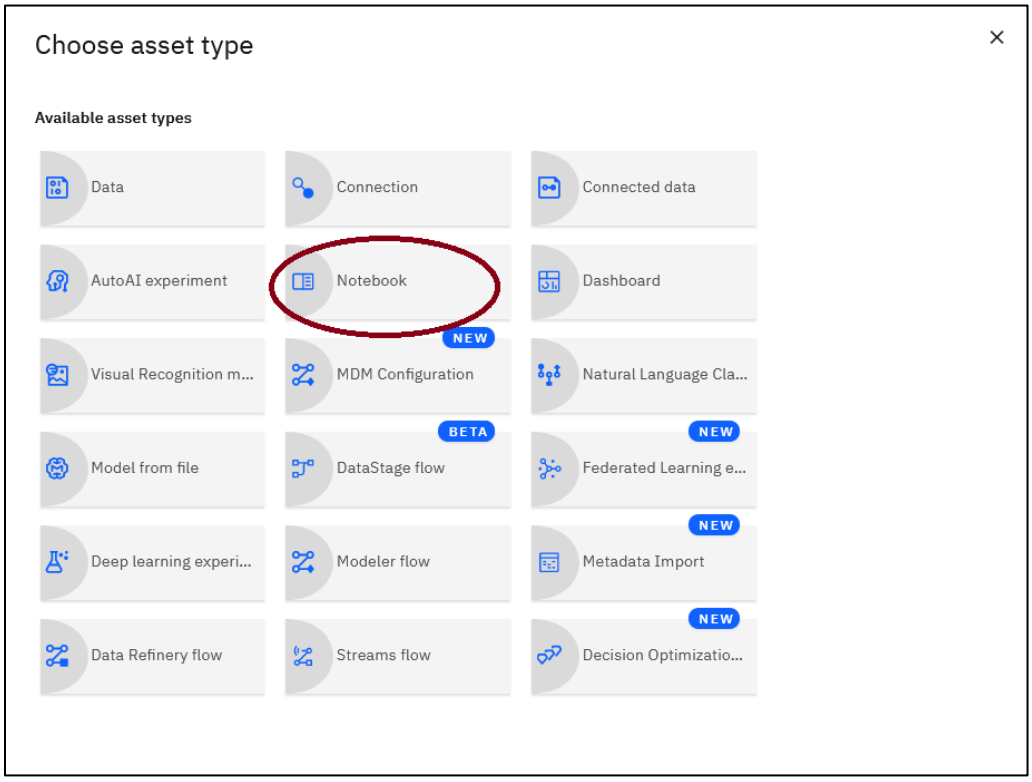
1. Click on the hamburger icon, then click on **Projects**, and then **Watson Studio Labs** (or whatever you named the project)



2. We are now going to create a notebook in our project. This notebook will be created from a url that points to the AIF notebook in the github repository. Click the **Add to project** link.



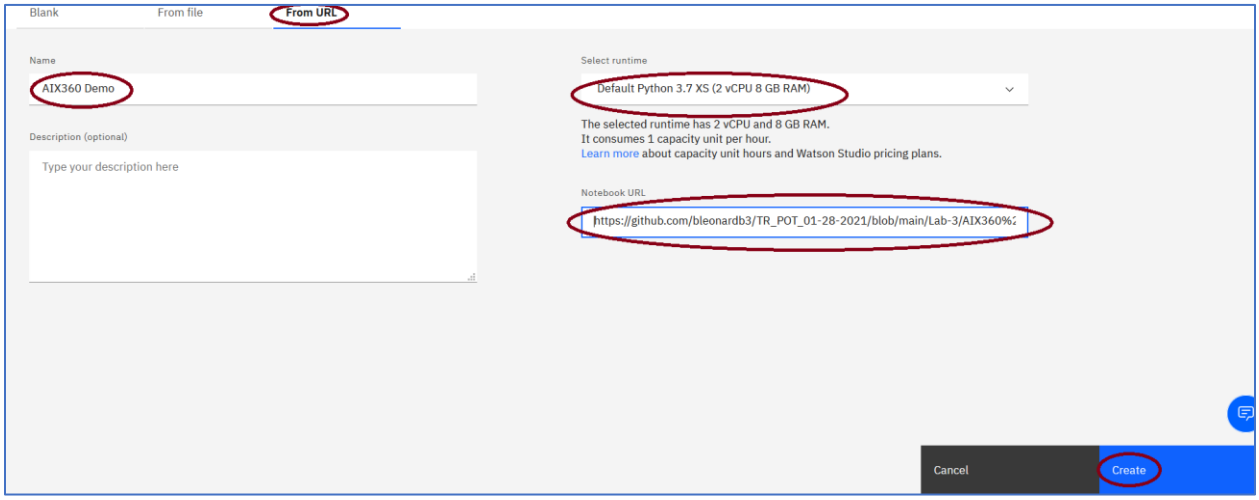
3. Click on **Notebook**



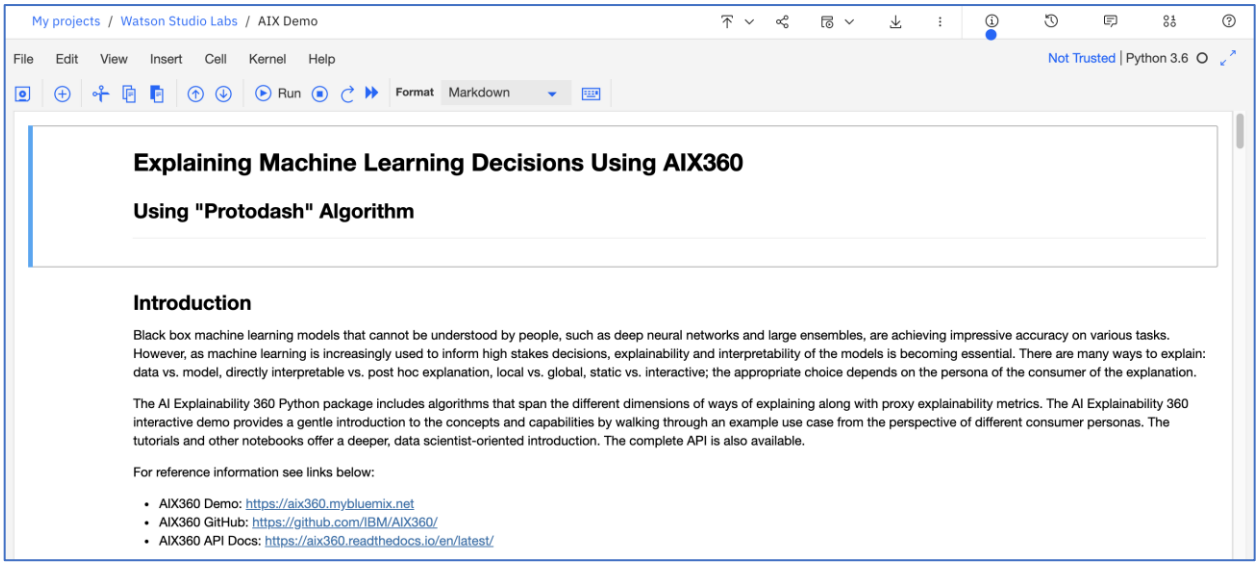
4. Click on **From URL** under **New Notebook,** enter **AIX360 Demo** for the **Name**, optionally enter a **Description**, leave the default for the **runtime**, and cut and paste the following url into the **Notebook URL** field.

https://github.com/bleonardb3/TR_POT_01-28-2021/blob/main/Lab-3/AIX360%20Demo.ipynb

Click **Create.**



5. Place the cursor in the first documentation cell.



6. Execute the cells in the notebook. Please go cell by cell and read the documentation or code comments to follow along. For those unfamiliar 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 the **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.