

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:

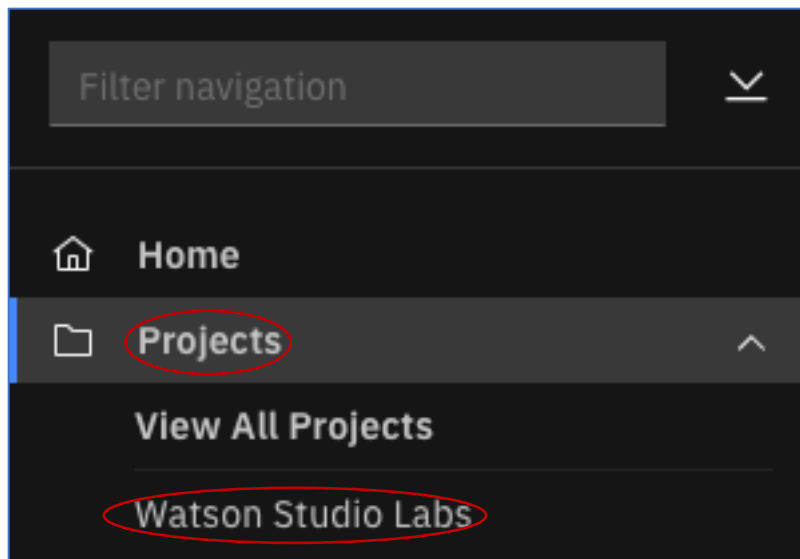
1. Load a dataset using a download link
2. Create, train, and evaluate a XGBoost model
3. Use Protodash Algorithm to extract similar examples and compare with the current patient's case

Lab Steps

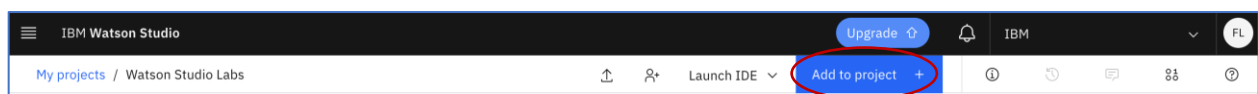
Step 1 - Create a Jupyter Notebook

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

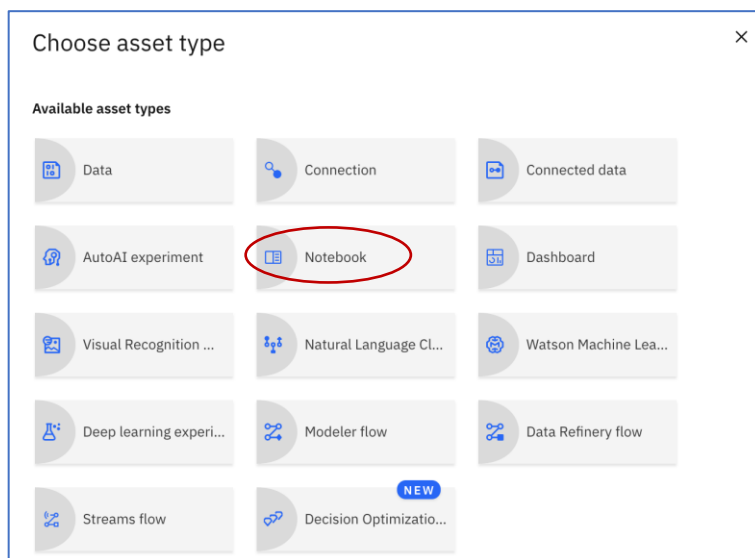




2. We are now going to create a notebook in our project. This notebook will be created from a url that points to the AIF notebook in the github repository. Click the **Add to project** link.



3. Click on **Notebook**



- 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_10-15-2020/blob/main/Lab-2/AIX360%20Demo.ipynb

Click **Create**.

Blank From file **From URL**

Name: **AIX360 Demo**

Description (optional):
Type your description here

Select runtime: Default Python 3.6 XS (2 vCPU 8 GB RAM)

The selected runtime has 2 vCPU and 8 GB RAM. It consumes 1 capacity unit per hour. [Learn more](#) about capacity unit hours and Watson Studio pricing plans.

Notebook URL: https://github.com/bleonardb3/TR_POT_10-15-2020/blob/main/Lab-2/AIX360%20Demo.ipynb

Cancel **Create**

- Place the cursor in the first documentation cell.

My projects / Watson Studio Labs / AIX Demo

File Edit View Insert Cell Kernel Help

Not Trusted | Python 3.6

Explaining Machine Learning Decisions Using AIX360

Using "Protodash" Algorithm

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 is becoming essential. There are many ways to explain: data vs. model, directly interpretable vs. post hoc explanation, local vs. global, static vs. interactive; the appropriate choice depends on the persona of the consumer of the explanation.

The AI Explainability 360 Python package includes algorithms that span the different dimensions of ways of explaining along with proxy explainability metrics. The AI Explainability 360 interactive demo provides a gentle introduction to the concepts and capabilities by walking through an example use case from the perspective of different consumer personas. The tutorials and other notebooks offer a deeper, data scientist-oriented introduction. The complete API is also available.

For reference 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/>

- Execute the code cells in the notebook. 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 **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.