

# Watson Studio : Machine Learning with SparkML

## Introduction

In this lab, we will explore machine learning using Spark ML. We will exploit Spark ML's high-level APIs built on top of DataFrames to create and tune machine learning pipelines. We will utilize Spark ML's feature transformers to convert, modify and scale the features that will be used to develop the machine learning model. Finally, we will evaluate and cross validate our model to demonstrate the process of determining a best fit model, load the results in the database, and save the model to the model repository.

We are using machine learning to try to predict records that a human has not seen or vetted before. We will use these predictions to sort the highest priority records for a human to look at. We will use as a training set for the algorithm simulated data that has been vetted by an analyst as high, medium or low.

## End-to-End Data Science

The general flow of the End to End Data Science PoT will be guided by the activities shown in Figure 1- End to End Flow. This lab spans the Prepare Data, Build Model, and Save and Deploy activities.

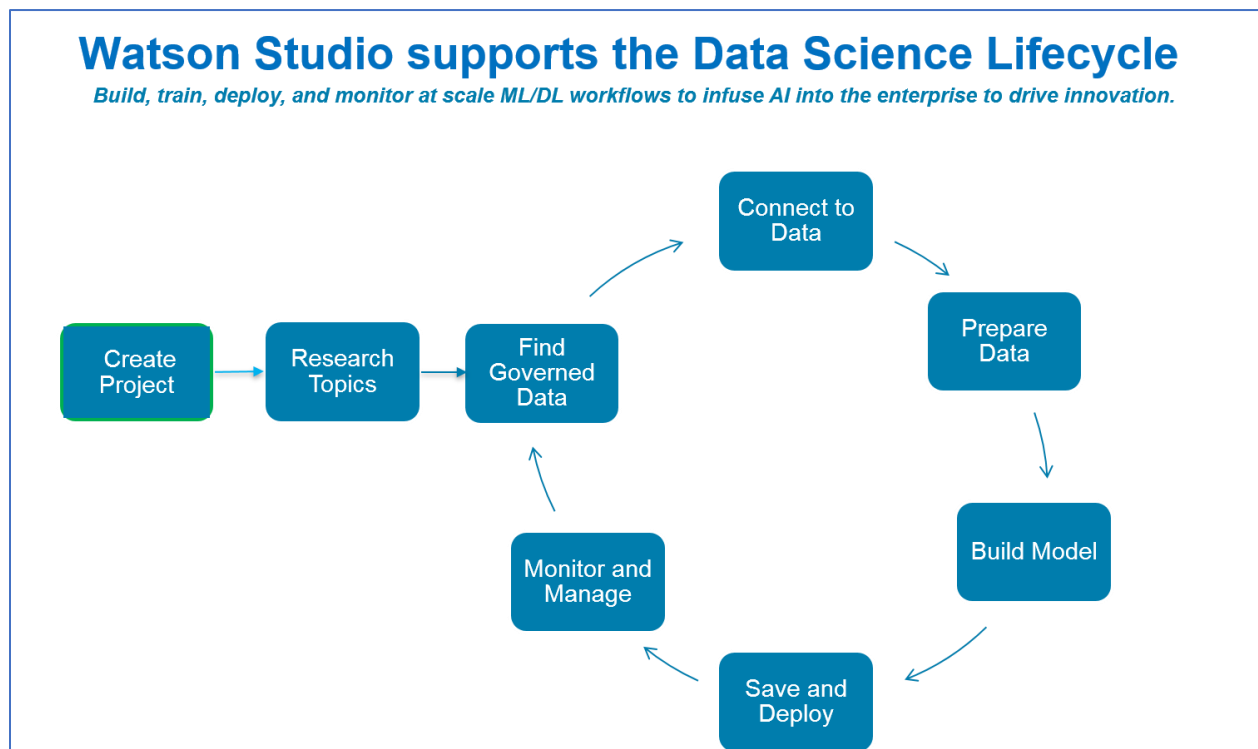


Figure 1- End to End Flow

## Objectives

Upon completing the lab, you will know how to:

- Join data from three sources.
- Identify labels and transform data.
- Conduct feature engineering for algorithm data.
- Declare a machine learning model.
- Setup the Pipeline for data transforms and training.
- Train the data.
- Show and evaluate machine learning results.
- Automatically tune machine learning results.
- Score data and load into a new DB2 table.
- Save the model to the model repository.

## Female Human Trafficking Data


The data sets used for this lab consist of **simulated** travel itinerary data. The use case corresponds to an analyst reviewing the travel data to assign a risk of trafficking. The risk is recorded as the VETTING\_LEVEL column in the dataset. Some of the records have already been analyzed and have a VETTING\_LEVEL of low (value is 30), medium (value is 20), or high risk (value is 10). Others have not yet been vetted (value is 100). We will use the data that has been vetted to train a model to predict the risk for the unvetted records. This can be used to automate the process and augment the analyst. For example, one option would be to send the predicted high-risk persons to the analyst for further investigation.

The OCCUPATION data included in the travel data is very granular. For modeling purposes, it was decided to categorize the OCCUPATION data. Two additional datasets are used for this purpose. The occupation.csv dataset maps the granular occupation data to a category code. The categories dataset maps a category code to a category description. These datasets will be joined to the main dataset to prepare the data for modeling.

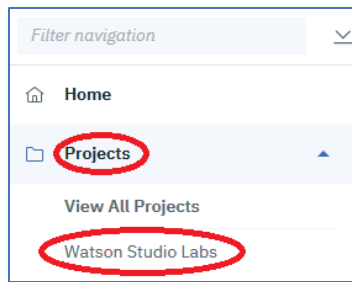
Other columns in the dataset are similarly very granular and could also be categorized for modeling purposes. This lab does not include steps to accomplish this, but it would be similar to what was done for the occupation column.

## Lab Steps

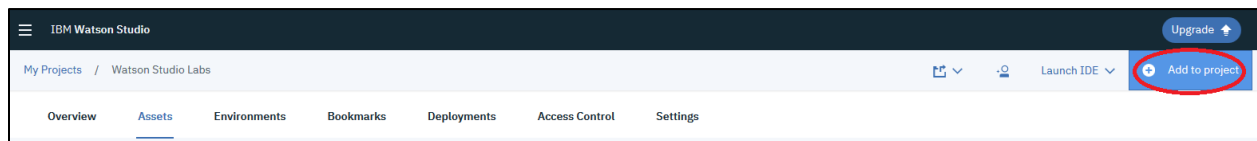
### Step 1 - Create a Jupyter Notebook

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

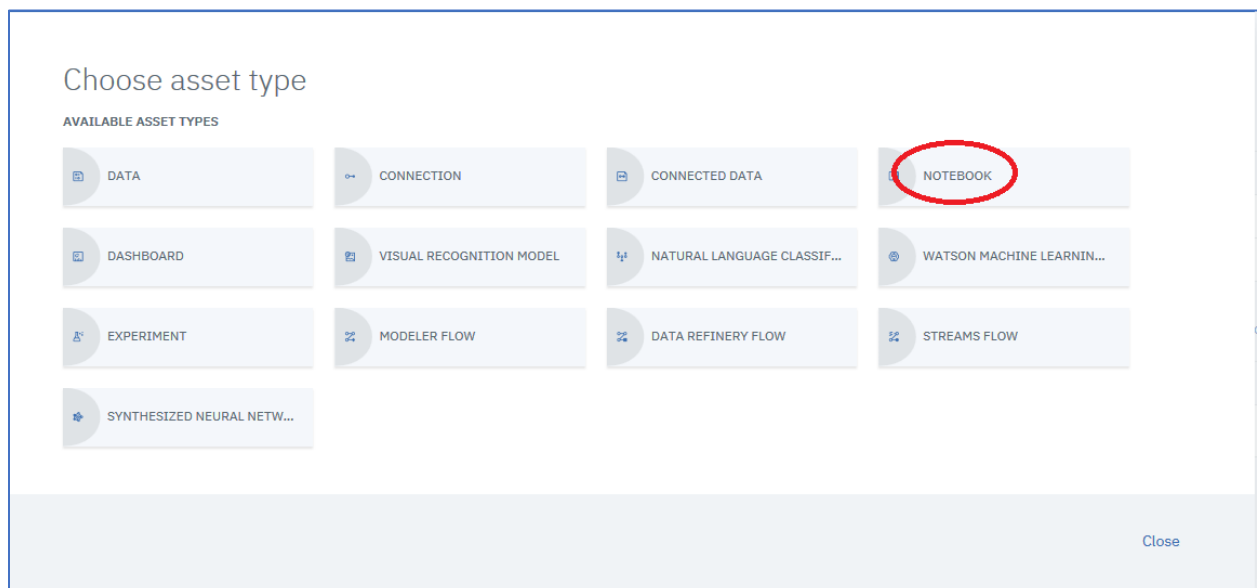




2. We are now going to create a notebook in our project. This notebook will be created from a url that points to the Machine Learning with SparkML notebook in the github repository. Click the **Add to project** link.



3. Click on **NOTEBOOK**



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

[https://github.com/bleonardb3/DS\\_POT\\_05-23/blob/master/Lab-5/Machine%20Learning%20with%20SparkML.ipynb](https://github.com/bleonardb3/DS_POT_05-23/blob/master/Lab-5/Machine%20Learning%20with%20SparkML.ipynb)

**Select the Runtime.**

**MAKE SURE TO SELECT Default Spark Python 3.6 XS (Driver with 1vCPU ...**

Click **Create Notebook**.

New notebook

Blank From file **From URL**

Name\*  
Machine Learning with SparkML 21 Characters Remaining

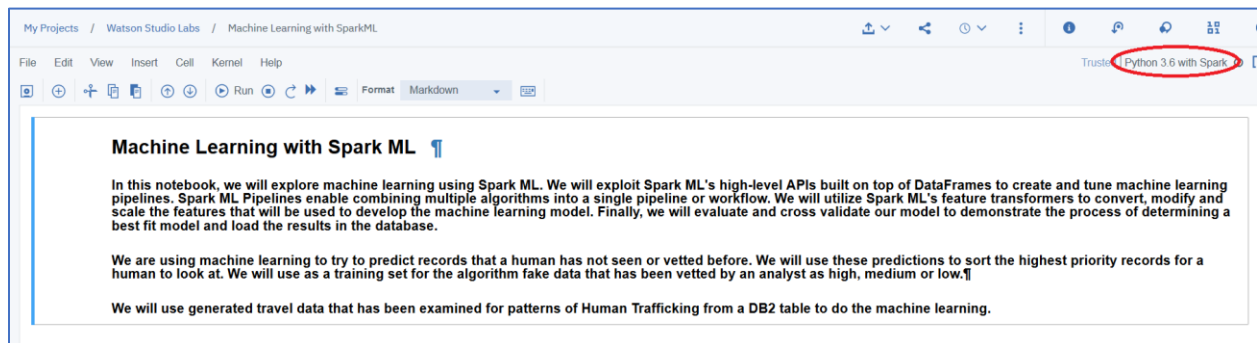
Description  
Type your Description here

Notebook URL\*  
https://github.com/bleonardb3/DS\_POT\_05-09/blob/master/Lab-5/Machine%20Learnin...

Select runtime\* Includes notebook environments ⓘ  
Default Spark Python 3.6 XS (Driver with 1 vCPU and 4 GB RAM, 2 executors with 1 v...  
The selected runtime uses one driver with 1 vCPU and 4 GB RAM, and 2 executors each with 1 vCPU and 4 GB RAM.  
This runtime consumes 1.5 capacity units per hour.  
Learn more about capacity unit hours and Watson Studio pricing plans.

Cancel **Create Notebook**

5. Please make sure the notebook has Python 3.6 with Spark in the top right corner.



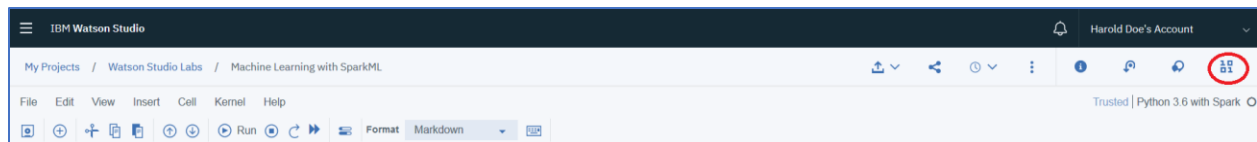
- A Jupyter notebook consists of a series of cells. These cells are of 3 types (1) documentation cells containing markdown, (2) code cells (denoted by a bracket on the left of the cell) where you write Python code, R, or Scala code depending on the type of notebook, and (3) output cells where the result of the code is placed. 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** 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.
- Before executing the cells in the notebook, we are going to use the IBM value-add code generator to insert code in 3 code cells that will read in the 3 input files and code in 1 code cell that will specify the database credentials. Scroll down in the notebook until you see the **Read Data Asset- female\_human\_trafficking** – See **Lab Instructions** and put

the cursor in the code cell underneath the comment lines. (Comments begin with the # sign).

Read Data Asset - female\_human\_trafficking - See Lab Instructions

```
In [ ]: # Insert SparkSession DataFrame code in this cell after the comments.  
# make CERTAIN to rename the default dataframe name (df_data_1 or df_data_2 or df_data_3, etc) to trafficking_df  
# Put cursor on the next line to Insert to code.
```

8. Click on the 1/0 icon.  at the top right.



9. Click on the insert to code down arrow  below **female human trafficking** and click on **insertSparkSession DataFrame**.

