## 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.

Diagram

Description automatically generated

Figure 1- End to End Flow

# Objectives

Upon completing the lab, you will know how to:

* Create a project token
* 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 model.
* Evaluate and show model results
* Automatically tune model
* 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 project token

The project token will be used to invoke project APIs.

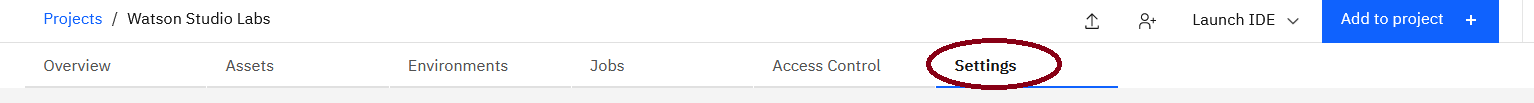
1. Click on the hamburger icon , then click on **Projects**, and then **Watson Studio Labs** (or whatever you named the project)



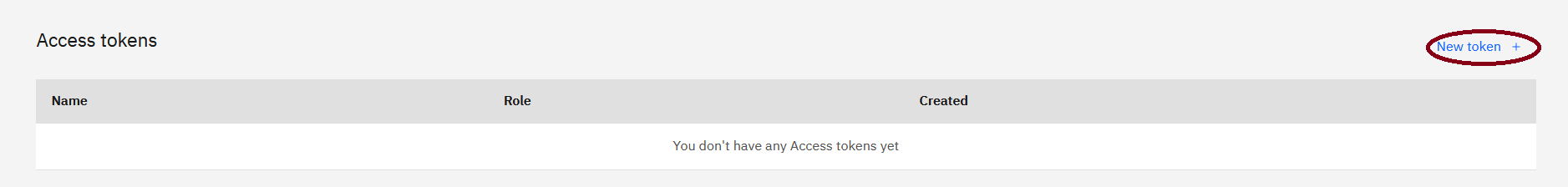
Graphical user interface, application

Description automatically generated

1. Click on **Settings**



1. Scroll down and click on **New** **token**.



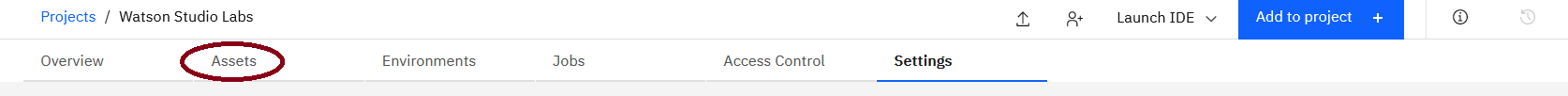
1. Enter **Watson** **Studio** **Labs** for the **Name**, select **Editor** for the **Access** **role**, and click **Create**.

Graphical user interface, text, application

Description automatically generated

## Step 2 - Create a Jupyter Notebook

1. Click on the **Assets** tab.



1. 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.

Graphical user interface, application

Description automatically generated

1. Click on **NOTEBOOK**

Graphical user interface, application

Description automatically generated

1. Click on **From URL** under New Notebook, enter **Machine Learning with SparkML** for the **Name**, and optionally enter a **Description**.

Select the Runtime. You will need to select a runtime that includes Python and Spark. MAKE SURE TO SELECT**Default Spark 3.0 & Python 3.7 (Driver: 1vCPU 4GB RAM, 2 Executors: 1vCPU 4GB RAM)**

**Graphical user interface, application, table

Description automatically generated**

1. Cut and paste the following url into the **Notebook URL** field

https://github.com/bleonardb3/DS\_POT\_01-14-2021/blob/master/Lab-5/Machine%20Learning%20with%20SparkML.ipynb

### Click Create Notebook.

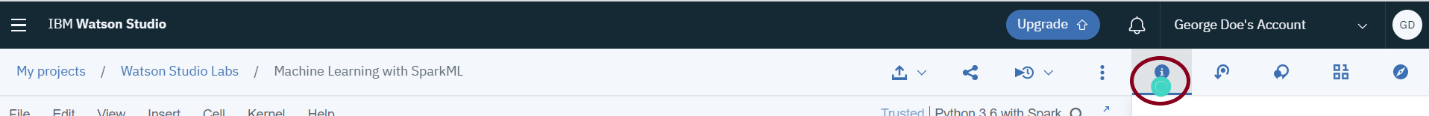
### Graphical user interface, application Description automatically generated

1. You should see the following notebook display.

A screenshot of a social media post

Description automatically generated

1. To verify the environment is correct, click on the notebook info icon 



1. Click on **Environment**.

A screenshot of a cell phone

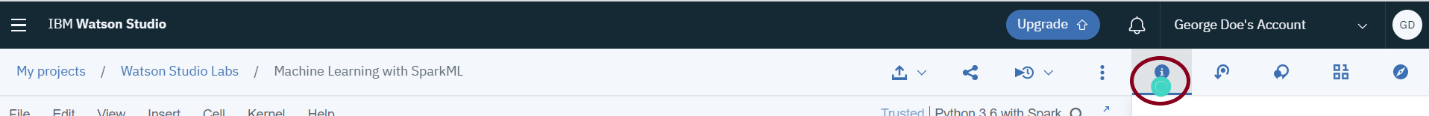
Description automatically generated

1. Verify that the language is Python 3.7 and the Spark version is 3.0.

Graphical user interface, text, application

Description automatically generated

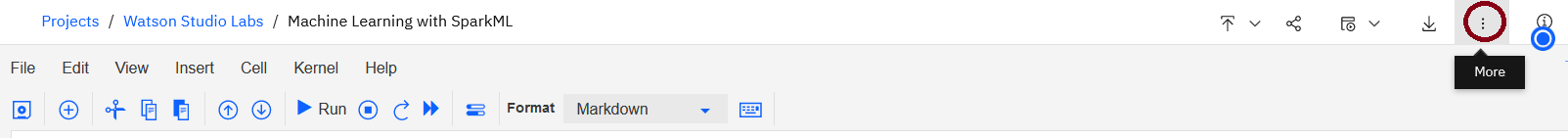
1. Click on the notebook info icon  to remove the Info panel.



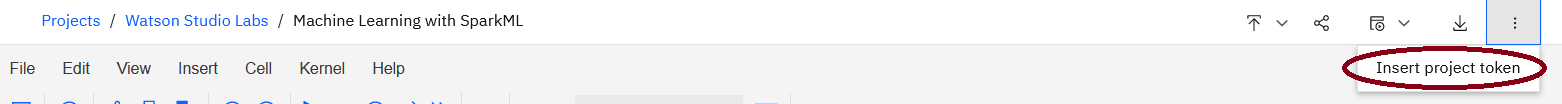
1. A Jupyter notebook consists of a series of cells. These cells are of 2 types (1) documentation cells containing markdown, and (2) code cells (denoted by a bracket on the left of the cell) where you write Python code, R, or Scala code depending on the type of notebook. Code cells can be run by putting 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.

## Step 3: Insert Project Token

1. Click on the vertical ellipse  at the top right of the Notebook.



1. Click **Insert** **project** **token**.



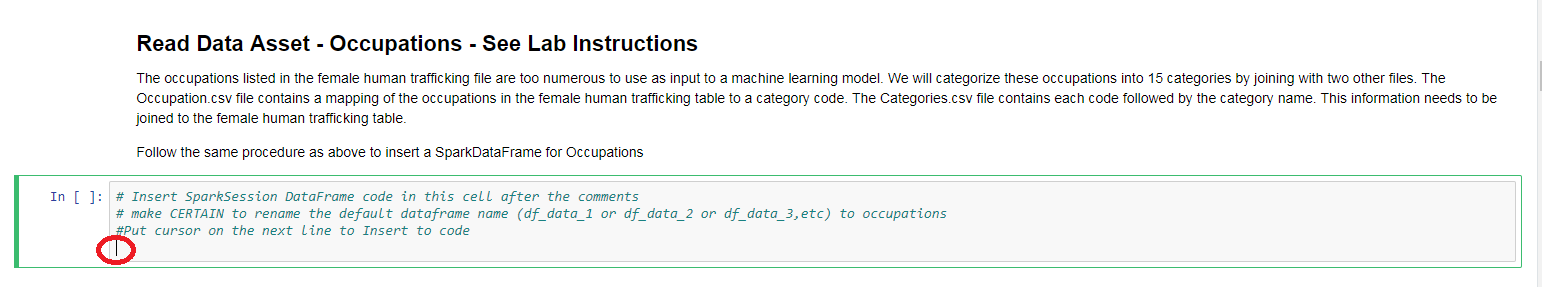
1. A project context is setup and the project id (in blue below), and project token (in green below) are shown below. The project id will be used later in the lab.

Graphical user interface, application

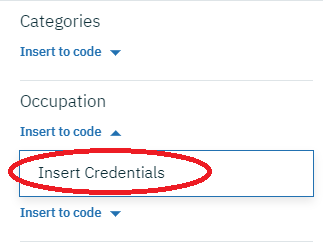
Description automatically generated

## Step 4: Insert Generated Code

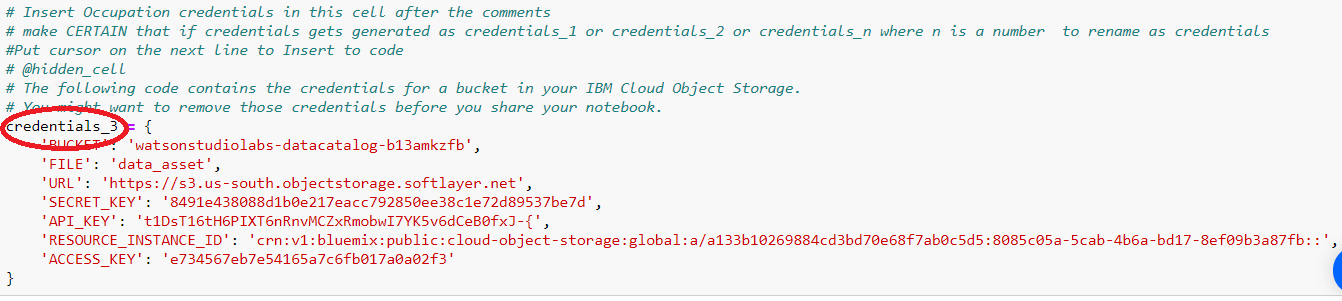
1. Before executing the cells in the notebook, we are going to use the IBM value-add code generator to insert code in 3 code cells. Scroll down in the notebook to **Read Data Asset – Occupations – See Lab Instructions.** Click the cursor underneath the commented lines in the code cell.



1. Click on the 1/0 icon.  at the top right.
2. Click on down arrow underneath **Occupation**. Click on **Insert Credentials.**



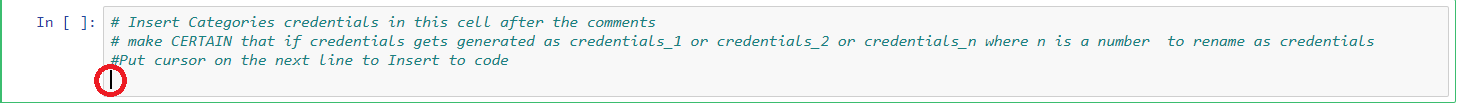
1. Locate the variable **credentials**. Make sure the variable does not have a number appended (e.g. **credentials\_1** or **credentials\_2** or **credentials\_3**, etc). If it does, change the variable to be **credentials** (without a number).



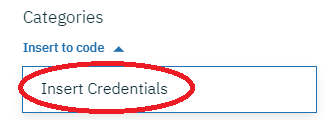
Change to:



1. Scroll down to **Read Data Asset – Categories – See Lab Instructions.** Click cursor underneath the commented lines in the code cell.



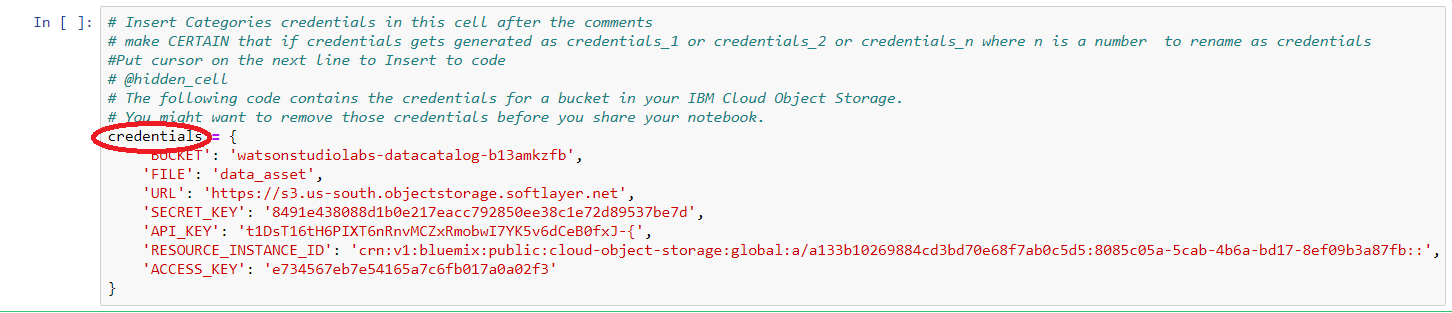
1. Click on down arrow underneath **Categories**. Click on **Insert Credentials.**



1. Locate the variable **credentials**. Make sure the variable does not have a number appended (e.g. **credentials\_1** or **credentials\_2** or **credentials\_3**, etc). If it does, change the variable to be **credentials** (without a number).



Change to:



## Step 5: Generate API Key to interface to Watson Machine Learning Service

In the notebook we will save our model to the Watson Machine Learning service model repository. We will need to generate an api key and determine the location of the Watson Machine Learning service. The procedure to obtain an api key and the location is described below.

1. Right-click on **IBM Cloud Pak for Data**, and the click on **Open Link in New Tab**.

Graphical user interface, text, application

Description automatically generated

1. Click on the new **IBM Cloud Pak for Data** browser tab.



1. Click on the hamburger icon , and then scroll down to the bottom to click on **IBM Cloud**.



A picture containing text

Description automatically generated

1. Click on **Dashboard**

Graphical user interface, application

Description automatically generated

1. Click on the IBM Command Shell icon .



1. 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**.

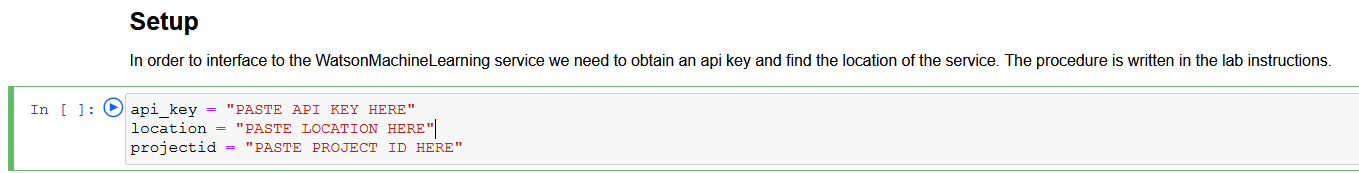
The location of the WatsonMachineLearning instance is shown highlighted in **blue** below. My instance location is shown to be in us-south.

Text

Description automatically generated

1. Copy and paste your api key and location values into the notebook cell below **Setup**. Click on the Machine Learning with SparkML to get back to the notebook.

Change:



The screen below shows the api key and location for my setup. Use your api key and location values.

To:

Graphical user interface, text, application

Description automatically generated

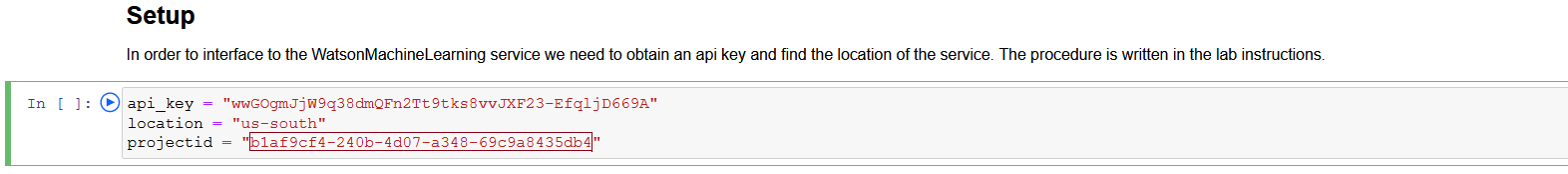
1. Copy and paste the project id from the first notebook cell, to the projectid variable.

Copy:



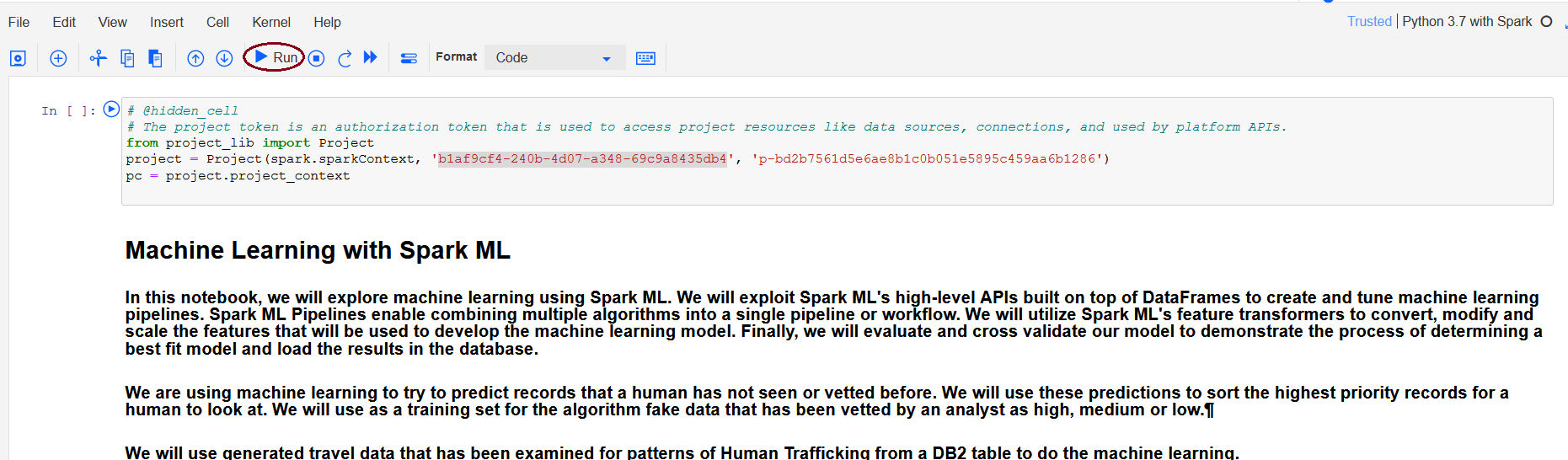
The screen below shows the project id from my setup. Copy your project id.

To:

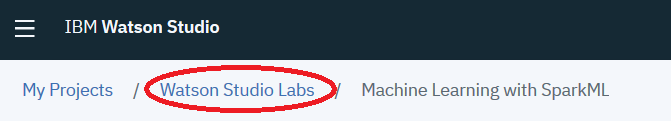


## Step 6: Execute the code cells in the notebook

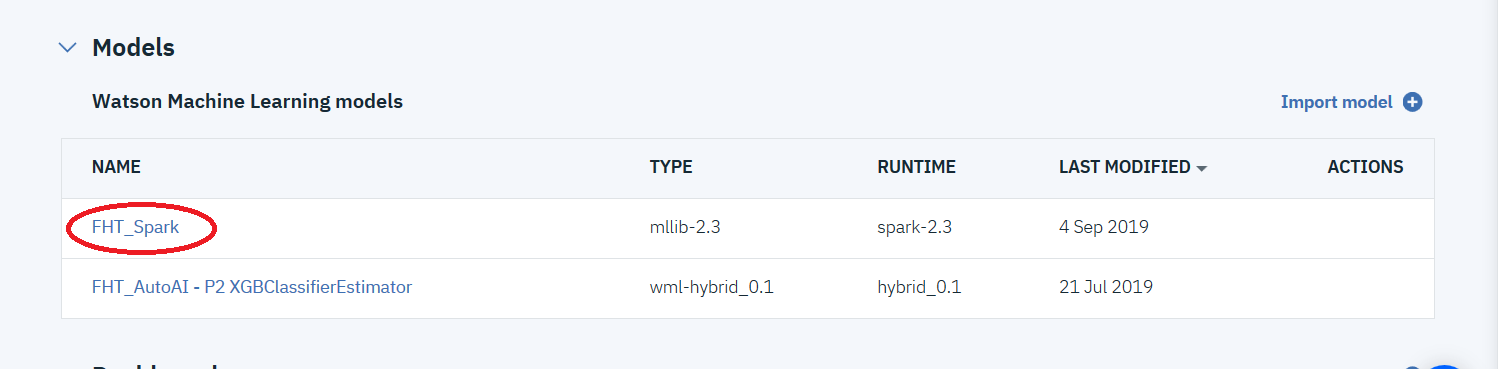
1. Scroll back to the top of the notebook. Click in to the first cell. Execute each of the code cells in order by clicking into each code cell starting at the top and pressing the **<Shift><Enter>** keys or by clicking into the first code cell and using the Run icon in the menu bar at the top. Read the 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 Step 2 below.**



1. Type **Ctrl-S** to save the notebook. Exit out of the notebook by clicking on the Watson Studio Labs in the breadcrumb area.



1. Scroll down the **Assets** page until you see the **Models** heading. The model listed was generated programmatically from the notebook using the Watson Machine Learning APIs. You should see the **FHT\_Spark** model in the list of Model Assets.



**You have completed Lab-5!**

* Create a project token
* Joined data from three sources.
* Identified labels and transformed data.
* Conducted feature engineering for algorithm data.
* Declared a machine learning model.
* Created the Pipeline for data transforms and training.
* Trained the model.
* Evaluated and showed model results.
* Automatically tuned model.
* Scored data and loaded into a new DB2 table.
* Saved the model to the model repository.