Watson Studio: Machine Learning with SparkML

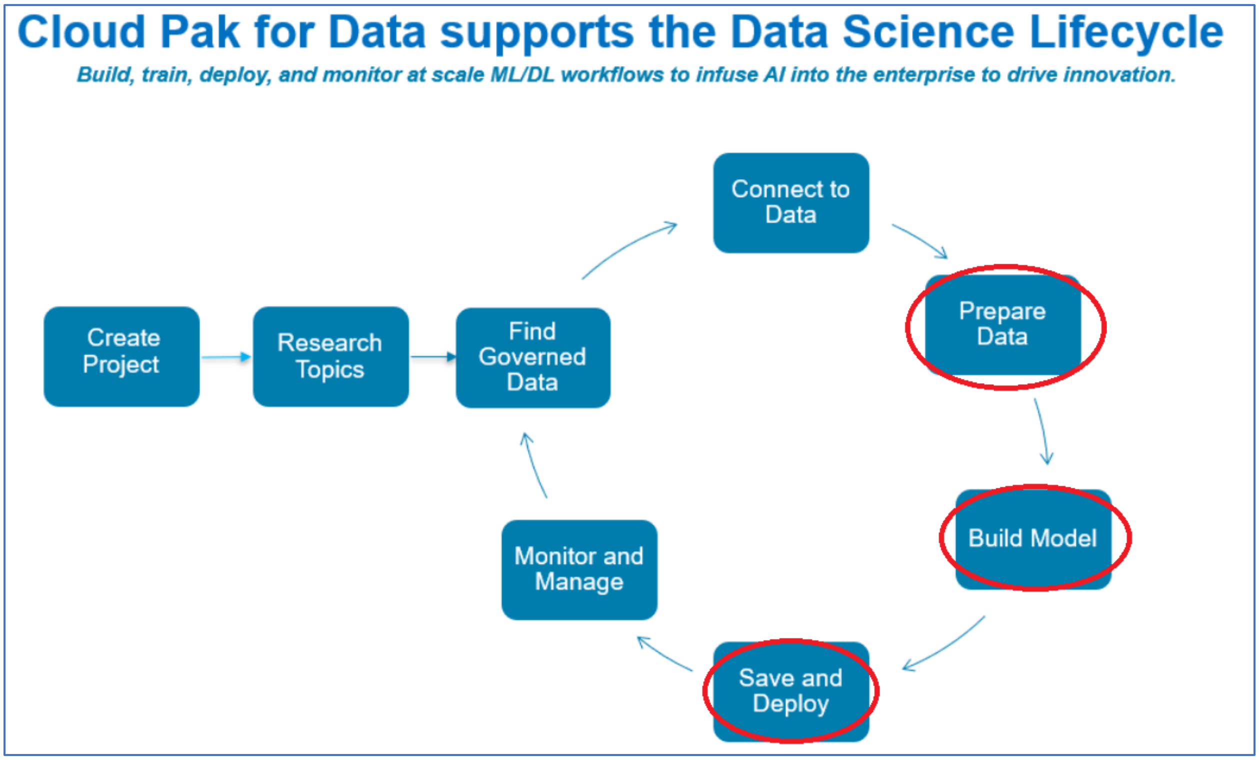
Introduction

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.



Cloud Pak for Data supports the Data Science Lifecycle

Build, train, deploy, and monitor at scale ML/DL workflows to infuse AI into the enterprise to drive innovation.

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.

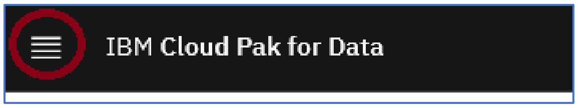
Other columns in the dataset are similarly very granular and could also be categorized for modeling purposes. This lab does not include steps to accomplish this, but it would be similar to what was done for the occupation column.

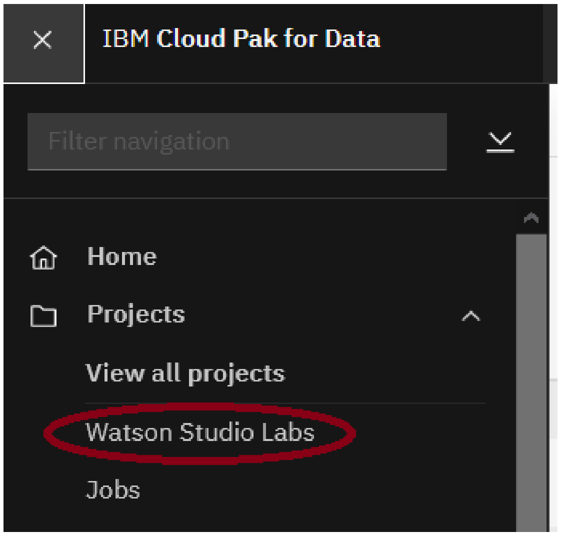
Lab Steps

Step 1 – Create a project token

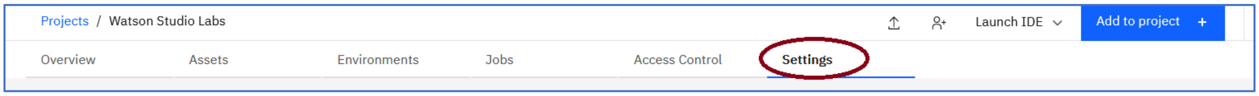
The project token will be used to invoke project APIs.

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

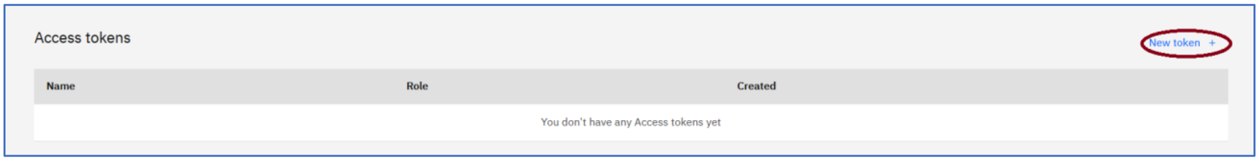




2. Click on **Settings**

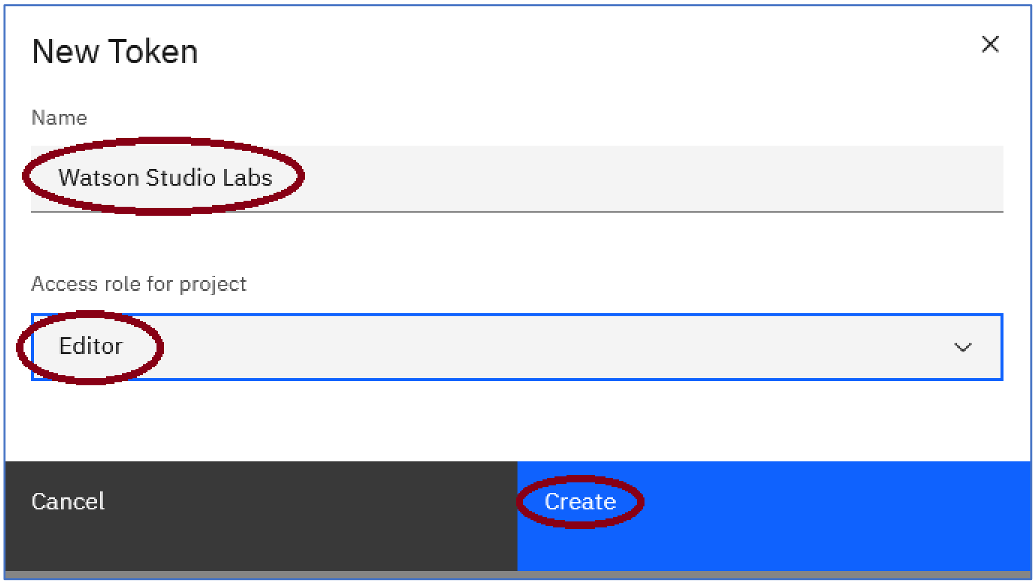


3. Scroll down and click on **New token**.


Access tokens

Name	Role	Created
You don't have any Access tokens yet		

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


New Token

Name

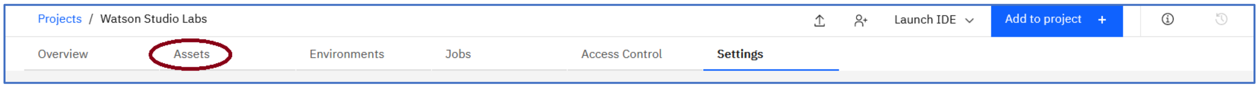
Watson Studio Labs

Access role for project

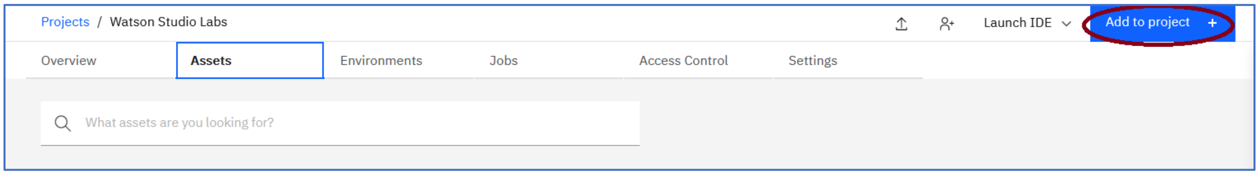
Editor

Step 2 - Create a Jupyter Notebook

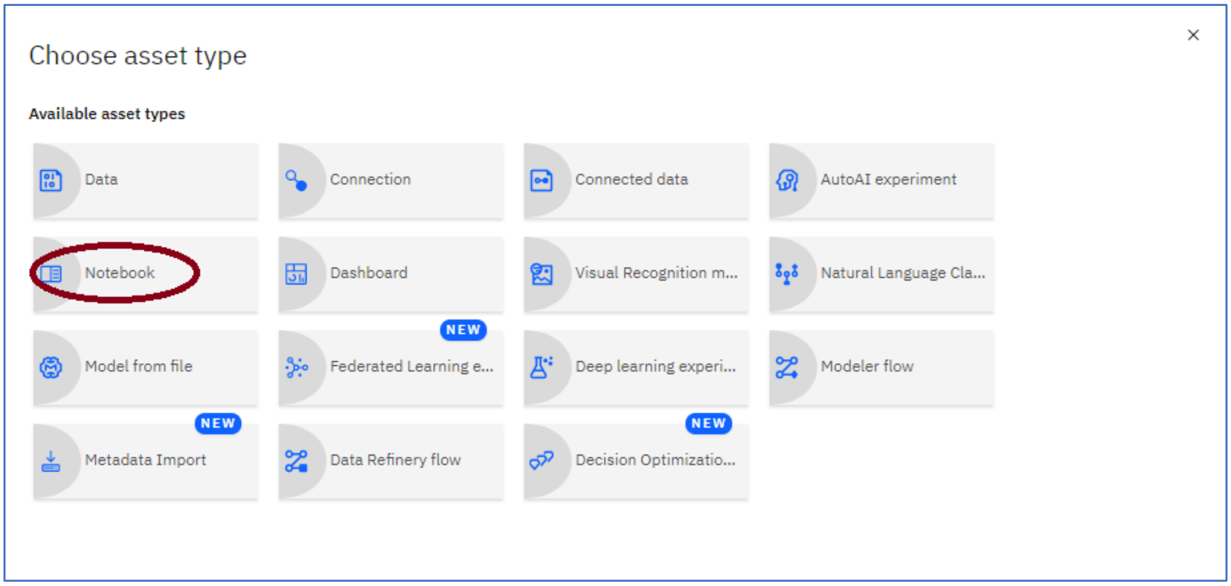
1. Click on the **Assets** tab.



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.

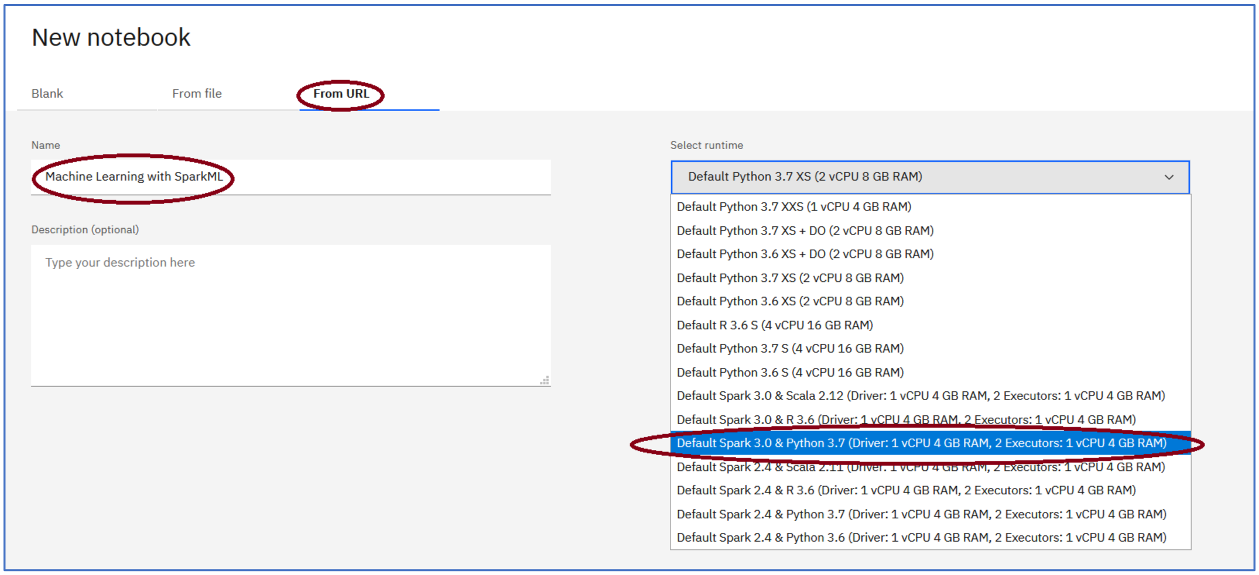


3. Click on **NOTEBOOK**



4. Click on **From URL** under New Notebook, enter **Machine Learning with SparkML** for the **Name**, 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)**



New notebook

Name: Machine Learning with SparkML

Description (optional): Type your description here

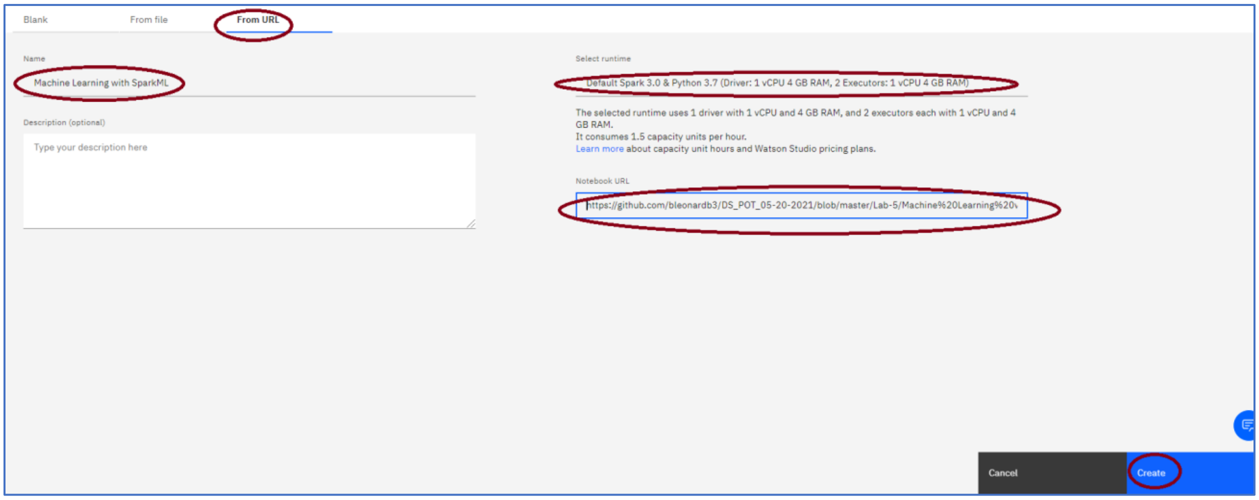
Select runtime: Default Python 3.7 XS (2 vCPU 8 GB RAM)

- Default Python 3.7 XXS (1 vCPU 4 GB RAM)
- Default Python 3.7 XS + DO (2 vCPU 8 GB RAM)
- Default Python 3.6 XS + DO (2 vCPU 8 GB RAM)
- Default Python 3.7 XS (2 vCPU 8 GB RAM)
- Default Python 3.6 XS (2 vCPU 8 GB RAM)
- Default R 3.6 S (4 vCPU 16 GB RAM)
- Default Python 3.7 S (4 vCPU 16 GB RAM)
- Default Python 3.6 S (4 vCPU 16 GB RAM)
- Default Spark 3.0 & Scala 2.12 (Driver: 1 vCPU 4 GB RAM, 2 Executors: 1 vCPU 4 GB RAM)
- Default Spark 3.0 & R 3.6 (Driver: 1 vCPU 4 GB RAM, 2 Executors: 1 vCPU 4 GB RAM)
- Default Spark 3.0 & Python 3.7 (Driver: 1 vCPU 4 GB RAM, 2 Executors: 1 vCPU 4 GB RAM)
- Default Spark 2.4 & Scala 2.11 (Driver: 1 vCPU 4 GB RAM, 2 Executors: 1 vCPU 4 GB RAM)
- Default Spark 2.4 & R 3.6 (Driver: 1 vCPU 4 GB RAM, 2 Executors: 1 vCPU 4 GB RAM)
- Default Spark 2.4 & Python 3.7 (Driver: 1 vCPU 4 GB RAM, 2 Executors: 1 vCPU 4 GB RAM)
- Default Spark 2.4 & Python 3.6 (Driver: 1 vCPU 4 GB RAM, 2 Executors: 1 vCPU 4 GB RAM)

5. Copy and paste the following url into the **Notebook URL** field

https://github.com/bleonardb3/DS_POT_05-20-2021/blob/master/Lab-5/Machine%20Learning%20with%20SparkML.ipynb

Click **Create Notebook.**



Name: Machine Learning with SparkML

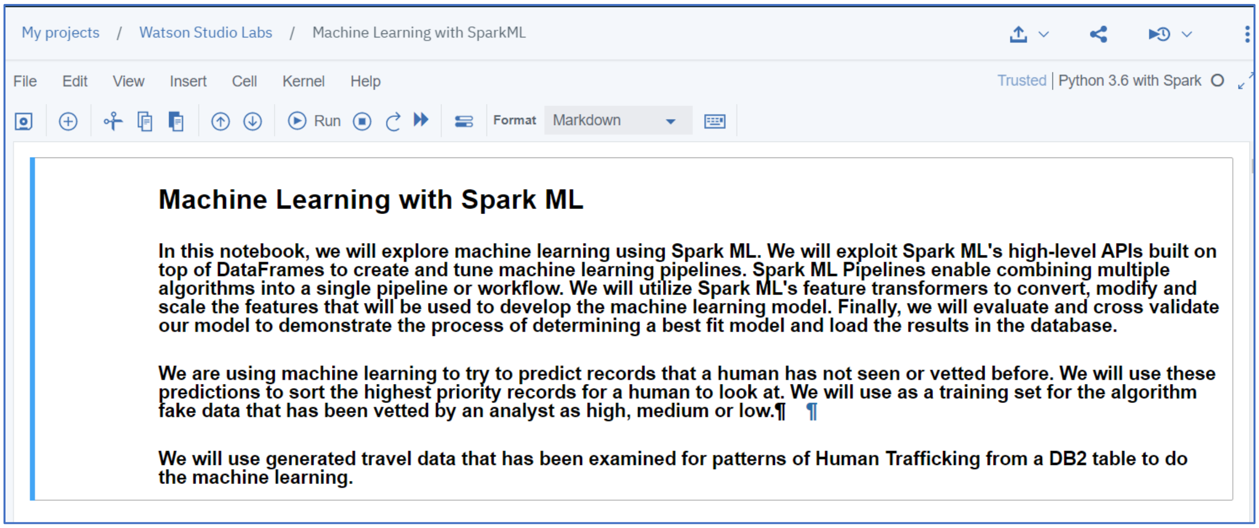
Description (optional): Type your description here

Select runtime: Default Spark 3.0 & Python 3.7 (Driver: 1 vCPU 4 GB RAM, 2 Executors: 1 vCPU 4 GB RAM)

The selected runtime uses 1 driver with 1 vCPU and 4 GB RAM, and 2 executors each with 1 vCPU and 4 GB RAM.
It consumes 1.5 capacity units per hour.
Learn more about capacity unit hours and Watson Studio pricing plans.

Notebook URL: https://github.com/bleonardb3/DS_POT_05-20-2021/blob/master/Lab-5/Machine%20Learning%20\

6. You should see the following notebook display.


My projects / Watson Studio Labs / Machine Learning with SparkML
File
Edit
View
Insert
Cell
Kernel
Help
Trusted | Python 3.6 with Spark
Run
Format
Markdown

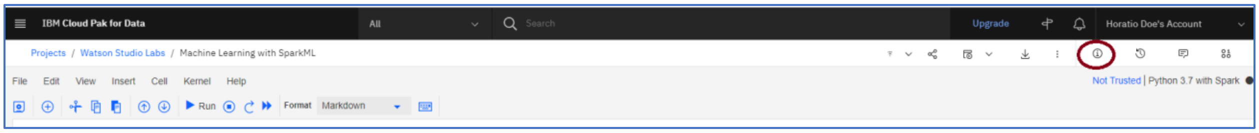
Machine Learning with Spark ML

In this notebook, 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. Spark ML Pipelines enable combining multiple algorithms into a single pipeline or workflow. 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 and load the results in the database.

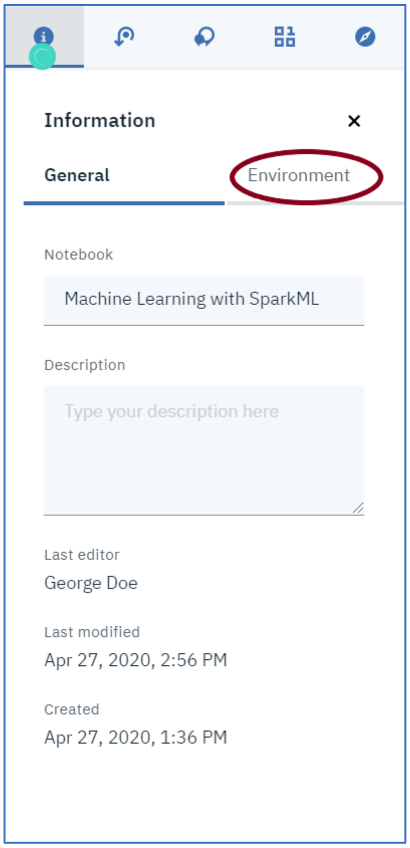
We are using machine learning to try to predict records that a human has not seen or vetted before. We will use these predictions to sort the highest priority records for a human to look at. We will use as a training set for the algorithm fake data that has been vetted by an analyst as high, medium or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

We will use generated travel data that has been examined for patterns of Human Trafficking from a DB2 table to do the machine learning.

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



8. Click on **Environment**.



Notebook
Machine Learning with SparkML

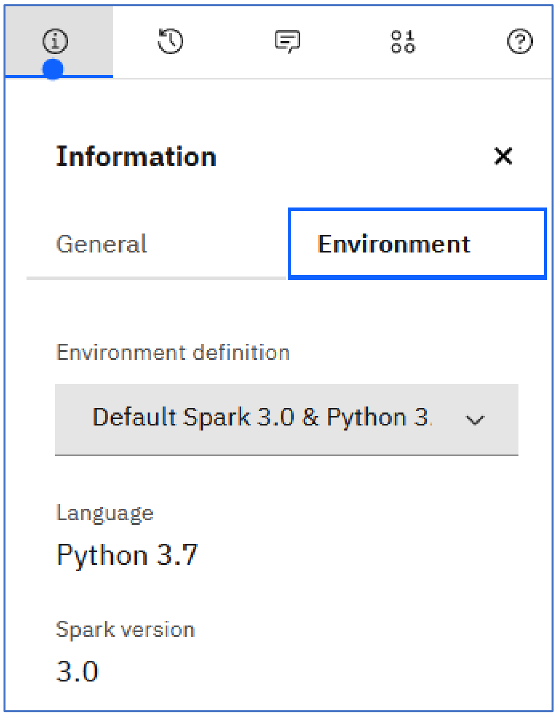
Description
Type your description here

Last editor
George Doe

Last modified
Apr 27, 2020, 2:56 PM

Created
Apr 27, 2020, 1:36 PM

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



Information

General | Environment

Environment definition

Default Spark 3.0 & Python 3

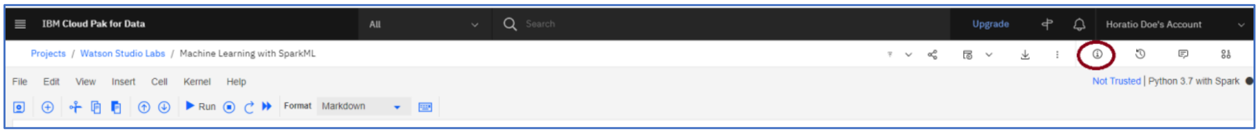
Language

Python 3.7

Spark version

3.0

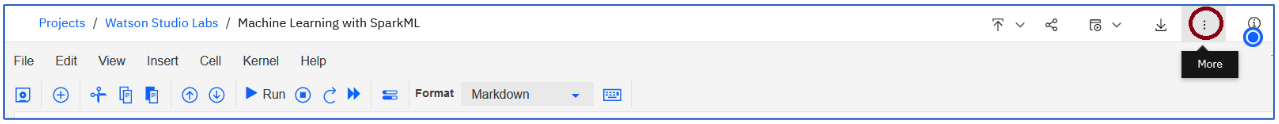
10. Click on the notebook info icon ⓘ to remove the Info panel.


IBM Cloud Pak for Data
All
Search
Upgrade
Horatio Doe's Account
Projects / Watson Studio Labs / Machine Learning with SparkML
File Edit View Insert Cell Kernel Help
Not Trusted | Python 3.7 with Spark
Run
Format
Markdown

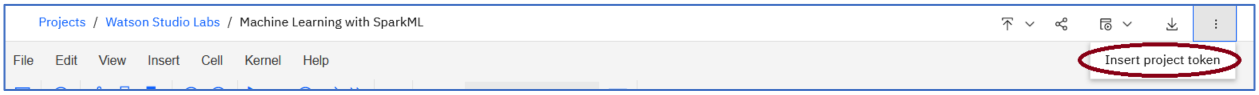
11. A Jupyter notebook consists of a series of cells. These cells are of 2 types (1) documentation cells containing markdown, and (2) code cells (denoted by a bracket on the left of the cell) where you write Python code, R, or Scala code depending on the type of notebook. Code cells can be run by putting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

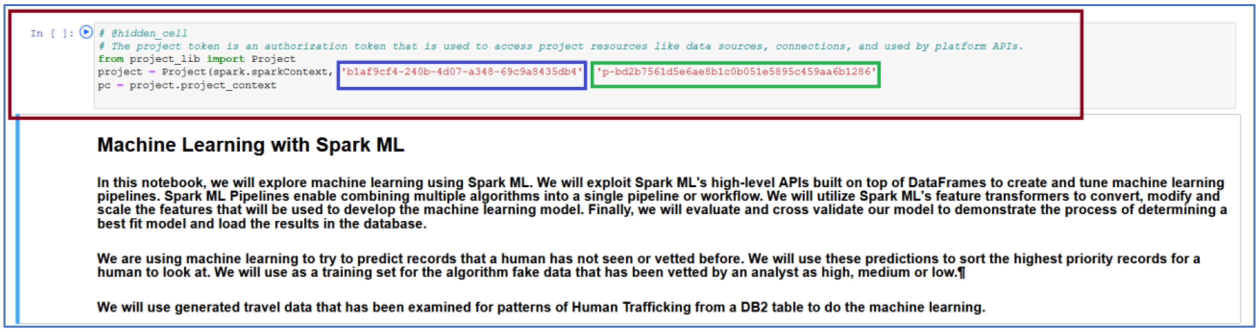
1. Click on the vertical ellipse ⋮ at the top right of the Notebook.


Projects / Watson Studio Labs / Machine Learning with SparkML
More
File Edit View Insert Cell Kernel Help
Run
Format Markdown

2. Click **Insert project token**.


Projects / Watson Studio Labs / Machine Learning with SparkML
File Edit View Insert Cell Kernel Help
Insert project token

3. 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.



```
In [ ]: # @hidden_cell
        # The project token is an authorization token that is used to access project resources like data sources, connections, and used by platform APIs.
        from project_lib import Project
        project = Project(spark.sparkContext, 'b1af9cf4-240b-4d07-a348-69c9a8435db4', 'p-bd2b7561d5e6ae8b1c0b051e5895c459aa6b1286')
        pc = project.project_context
```

Machine Learning with Spark ML

In this notebook, 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. Spark ML Pipelines enable combining multiple algorithms into a single pipeline or workflow. 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 and load the results in the database.

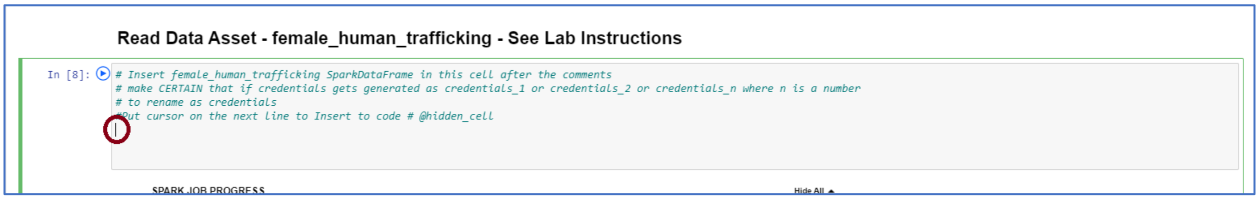
We are using machine learning to try to predict records that a human has not seen or vetted before. We will use these predictions to sort the highest priority records for a human to look at. We will use as a training set for the algorithm fake data that has been vetted by an analyst as high, medium or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

We will use generated travel data that has been examined for patterns of Human Trafficking from a DB2 table to do the machine learning.

Step 4: Insert Generated Code

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.

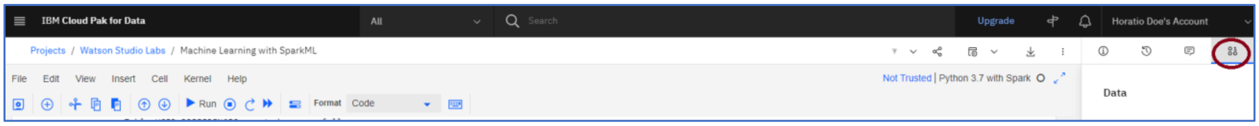
1. Scroll down in the notebook to **Read Data Asset - female_human_trafficking - See Lab Instructions.** Click the cursor underneath the commented lines in the code cell.



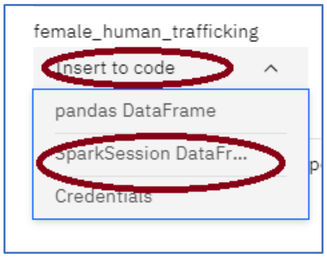
Read Data Asset - female_human_trafficking - See Lab Instructions

```
In [8]: # Insert female_human_trafficking SparkDataFrame in this cell after the comments
        # make CERTAIN that if credentials gets generated as credentials_1 or credentials_2 or credentials_n where n is a number
        # to rename as credentials
        #Put cursor on the next line to Insert to code # @hidden_cell
        |
```

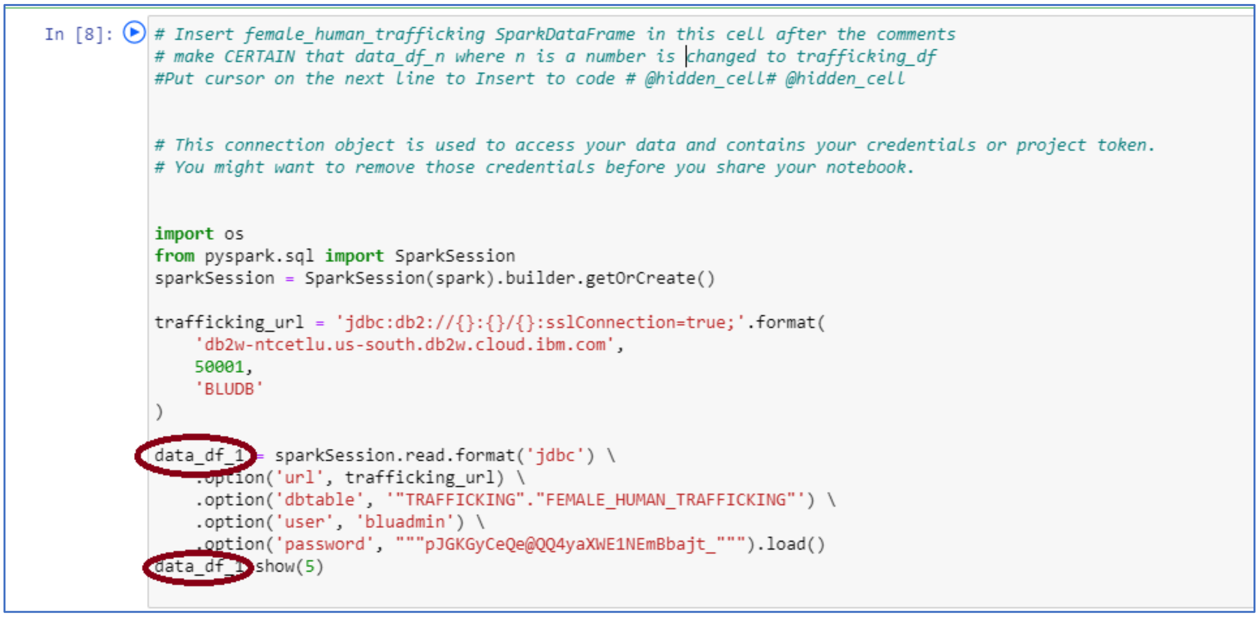
2. Click on the 1/0 icon. 10 01 at the top right.


IBM Cloud Pak for Data
All
Search
Upgrade
Horatio Doe's Account
Projects / Watson Studio Labs / Machine Learning with SparkML
File Edit View Insert Cell Kernel Help
Not Trusted | Python 3.7 with Spark
Run
Format Code
Data

3. Click on Insert to code ▾ down arrow underneath **female_human_trafficking**. Click on **Insert SparkDataFrame.**

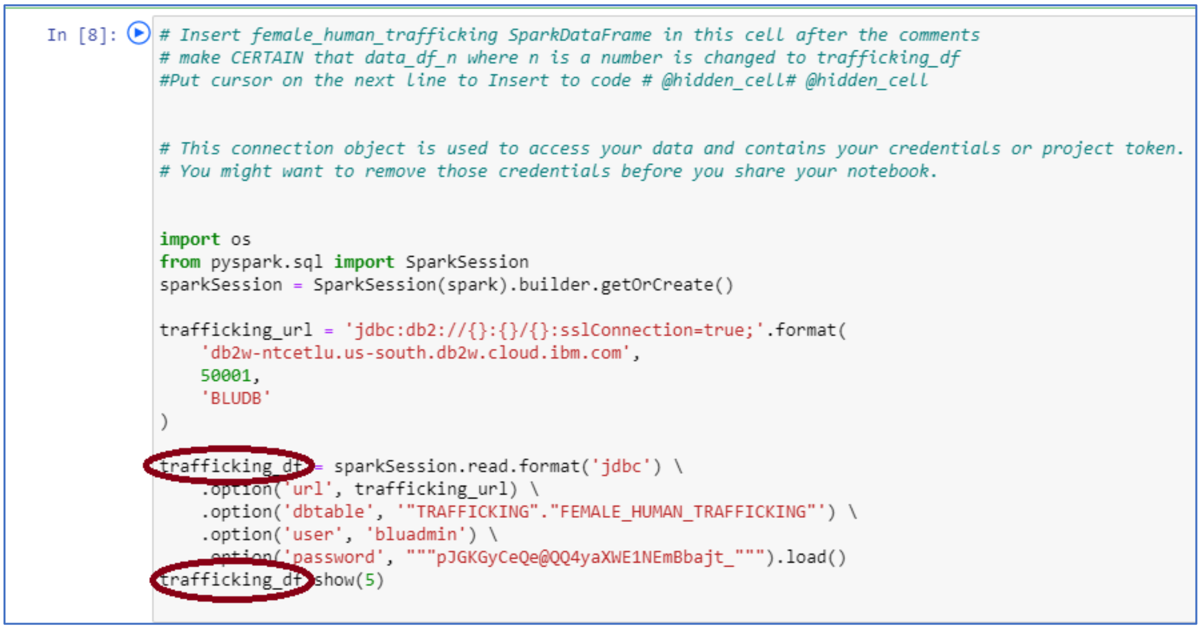


4. Locate the variable df_data_n, where “n” is a number. Make sure to change df_data_n to trafficking_df as below:



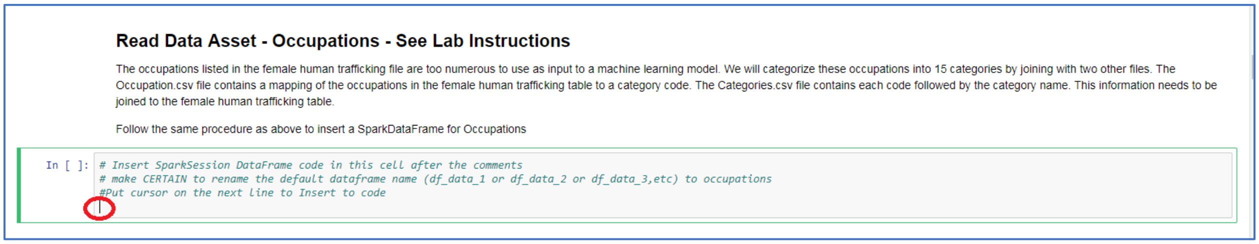
```
In [8]:  # Insert female_human_trafficking SparkDataFrame in this cell after the comments
         # make CERTAIN that data_df_n where n is a number is changed to trafficking_df
         #Put cursor on the next line to Insert to code # @hidden_cell# @hidden_cell


         # This connection object is used to access your data and contains your credentials or project token.
         # You might want to remove those credentials before you share your notebook.


         import os
         from pyspark.sql import SparkSession
         sparkSession = SparkSession(spark).builder.getOrCreate()

         trafficking_url = 'jdbc:db2://{}:{}/{}:sslConnection=true;'.format(
             'db2w-ntcetlu.us-south.db2w.cloud.ibm.com',
             50001,
             'BLUDB'
         )

         data_df_1 = sparkSession.read.format('jdbc') \
             .option('url', trafficking_url) \
             .option('dbtable', '"TRAFFICKING"."FEMALE_HUMAN_TRAFFICKING"') \
             .option('user', 'bluadmin') \
             .option('password', """pJGKGyCeQe@QQ4yaXWE1NEmBbajt_""").load()
         data_df_1.show(5)
```

Change to:



```
In [8]:  # Insert female_human_trafficking SparkDataFrame in this cell after the comments
         # make CERTAIN that data_df_n where n is a number is changed to trafficking_df
         #Put cursor on the next line to Insert to code # @hidden_cell# @hidden_cell


         # This connection object is used to access your data and contains your credentials or project token.
         # You might want to remove those credentials before you share your notebook.


         import os
         from pyspark.sql import SparkSession
         sparkSession = SparkSession(spark).builder.getOrCreate()

         trafficking_url = 'jdbc:db2://{}:{}/{}:sslConnection=true;'.format(
             'db2w-ntcetlu.us-south.db2w.cloud.ibm.com',
             50001,
             'BLUDB'
         )

         trafficking_df = sparkSession.read.format('jdbc') \
             .option('url', trafficking_url) \
             .option('dbtable', '"TRAFFICKING"."FEMALE_HUMAN_TRAFFICKING"') \
             .option('user', 'bluadmin') \
             .option('password', """pJGKGyCeQe@QQ4yaXWE1NEmBbajt_""").load()
         trafficking_df.show(5)
```

5. **Read Data Asset – Occupations – See Lab Instructions.** Click the cursor underneath the commented lines in the code cell.



Read Data Asset - Occupations - See Lab Instructions

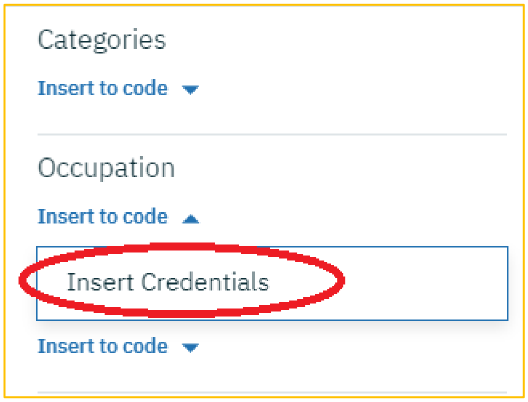
The occupations listed in the female human trafficking file are too numerous to use as input to a machine learning model. We will categorize these occupations into 15 categories by joining with two other files. The Occupation.csv file contains a mapping of the occupations in the female human trafficking table to a category code. The Categories.csv file contains each code followed by the category name. This information needs to be joined to the female human trafficking table.

Follow the same procedure as above to insert a SparkDataFrame for Occupations

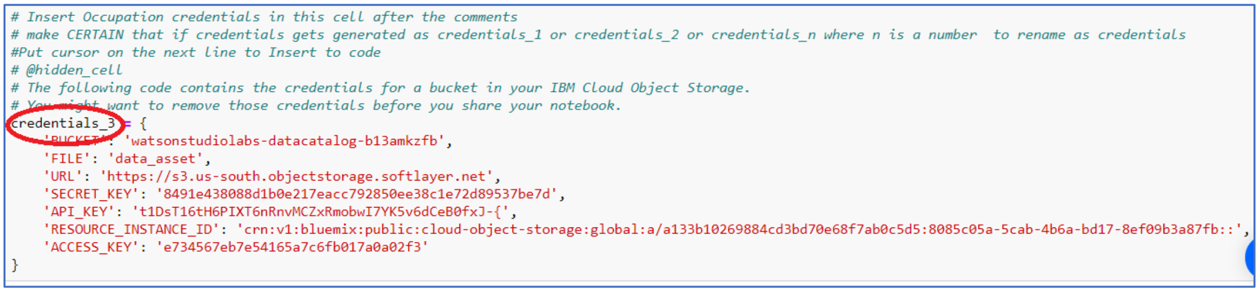
```
In [ ]: # Insert SparkSession DataFrame code in this cell after the comments
        # make CERTAIN to rename the default dataframe name (df_data_1 or df_data_2 or df_data_3,etc) to occupations
        #Put cursor on the next line to Insert to code
```

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

7. Click on Insert to code down arrow underneath **Occupation**. Click on **Insert Credentials.**

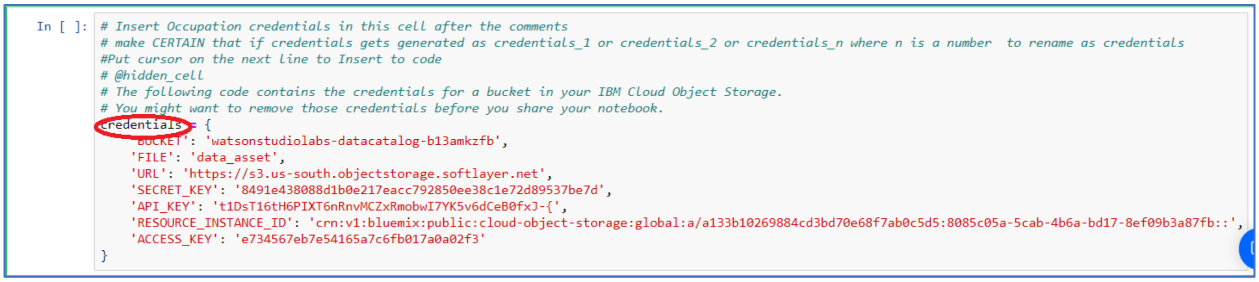


8. Locate the variable **credentials**. Make sure the variable does not have a number appended (e.g. **credentials_1** or **credentials_2** or **credentials_3**, etc). If it does, change the variable to be **credentials** (without a number).



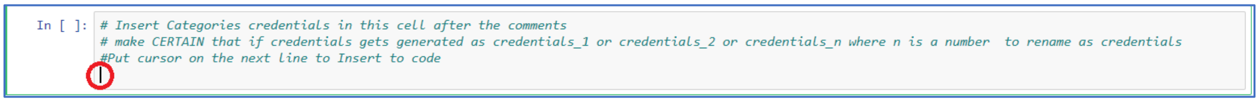
```
# Insert Occupation credentials in this cell after the comments
# make CERTAIN that if credentials gets generated as credentials_1 or credentials_2 or credentials_n where n is a number  to rename as credentials
#Put cursor on the next line to Insert to code
# @hidden_cell
# The following code contains the credentials for a bucket in your IBM Cloud Object Storage.
# You might want to remove those credentials before you share your notebook.
credentials_3 = {
    'BUCKET': 'watsonstudiolabs-datacatalog-b13amkzfb',
    'FILE': 'data_asset',
    'URL': 'https://s3.us-south.objectstorage.softlayer.net',
    'SECRET_KEY': '8491e438088d1b0e217eacc792850ee38c1e72d89537be7d',
    'API_KEY': 't1DsT16tH6PIXT6nRnvMCZxRmobwI7YK5v6dCeB0fxJ-{',
    'RESOURCE_INSTANCE_ID': 'crn:v1:bluemix:public:cloud-object-storage:global:a/a133b10269884cd3bd70e68f7ab0c5d5:8085c05a-5cab-4b6a-bd17-8ef09b3a87fb::',
    'ACCESS_KEY': 'e734567eb7e54165a7c6fb017a0a02f3'
}
```

Change to:

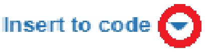


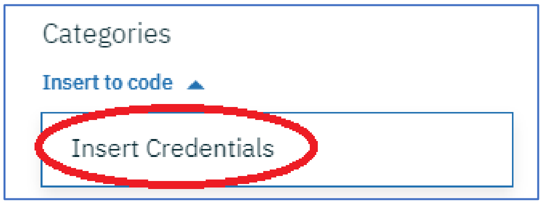
```
In [ ]: # Insert Occupation credentials in this cell after the comments
        # make CERTAIN that if credentials gets generated as credentials_1 or credentials_2 or credentials_n where n is a number  to rename as credentials
        #Put cursor on the next line to Insert to code
        # @hidden_cell
        # The following code contains the credentials for a bucket in your IBM Cloud Object Storage.
        # You might want to remove those credentials before you share your notebook.
        credentials = {
            'BUCKET': 'watsonstudiolabs-datacatalog-b13amkzfb',
            'FILE': 'data_asset',
            'URL': 'https://s3.us-south.objectstorage.softlayer.net',
            'SECRET_KEY': '8491e438088d1b0e217eacc792850ee38c1e72d89537be7d',
            'API_KEY': 't1DsT16tH6PIXT6nRnvMCZxRmobwI7YK5v6dCeB0fxJ-{',
            'RESOURCE_INSTANCE_ID': 'crn:v1:bluemix:public:cloud-object-storage:global:a/a133b10269884cd3bd70e68f7ab0c5d5:8085c05a-5cab-4b6a-bd17-8ef09b3a87fb::',
            'ACCESS_KEY': 'e734567eb7e54165a7c6fb017a0a02f3'
        }
```

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

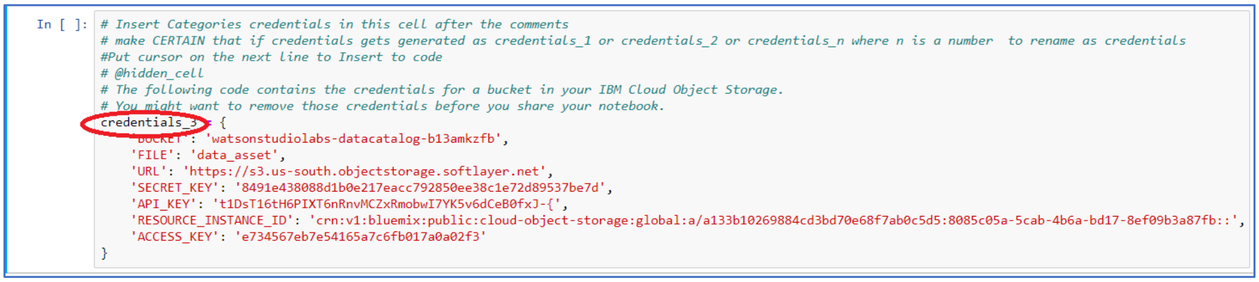


```
In [ ]: # Insert Categories credentials in this cell after the comments
        # make CERTAIN that if credentials gets generated as credentials_1 or credentials_2 or credentials_n where n is a number  to rename as credentials
        #Put cursor on the next line to Insert to code
        |
```

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

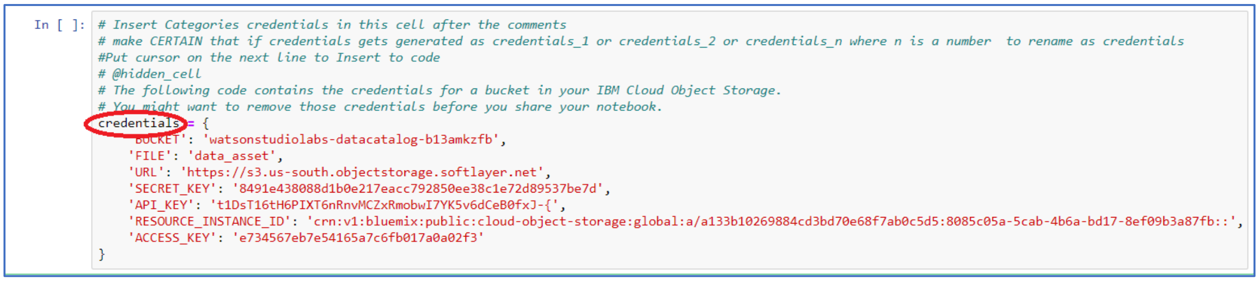


11. Locate the variable **credentials**. Make sure the variable does not have a number appended (e.g. **credentials_1** or **credentials_2** or **credentials_3**, etc). If it does, change the variable to be **credentials** (without a number).



```
In [ ]: # Insert Categories credentials in this cell after the comments
        # make CERTAIN that if credentials gets generated as credentials_1 or credentials_2 or credentials_n where n is a number  to rename as credentials
        #Put cursor on the next line to Insert to code
        # @hidden_cell
        # The following code contains the credentials for a bucket in your IBM Cloud Object Storage.
        # You might want to remove those credentials before you share your notebook.
        credentials_3 = {
            'BUCKET': 'watsonstudiolabs-datacatalog-b13amkzfb',
            'FILE': 'data_asset',
            'URL': 'https://s3.us-south.objectstorage.softlayer.net',
            'SECRET_KEY': '8491e438088d1b0e217eacc792850ee38c1e72d89537be7d',
            'API_KEY': 't1DsT16tH6PIXT6nRnvMCZxRmobwI7YK5v6dCeB0fxJ-{',
            'RESOURCE_INSTANCE_ID': 'crn:v1:bluemix:public:cloud-object-storage:global:a/a133b10269884cd3bd70e68f7ab0c5d5:8085c05a-5cab-4b6a-bd17-8ef09b3a87fb::',
            'ACCESS_KEY': 'e734567eb7e54165a7c6fb017a0a02f3'
        }
```

Change to:

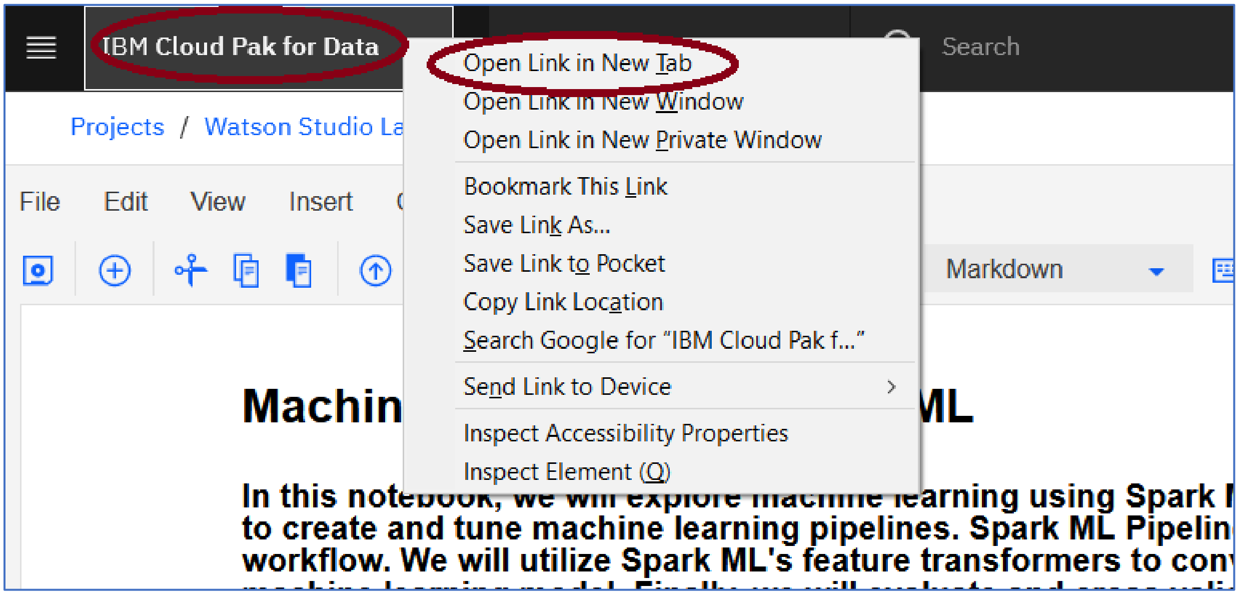


```
In [ ]: # Insert Categories credentials in this cell after the comments
        # make CERTAIN that if credentials gets generated as credentials_1 or credentials_2 or credentials_n where n is a number  to rename as credentials
        #Put cursor on the next line to Insert to code
        # @hidden_cell
        # The following code contains the credentials for a bucket in your IBM Cloud Object Storage.
        # You might want to remove those credentials before you share your notebook.
        credentials = {
            'BUCKET': 'watsonstudiolabs-datacatalog-b13amkzfb',
            'FILE': 'data_asset',
            'URL': 'https://s3.us-south.objectstorage.softlayer.net',
            'SECRET_KEY': '8491e438088d1b0e217eacc792850ee38c1e72d89537be7d',
            'API_KEY': 't1DsT16tH6PIXT6nRnvMCZxRmobwI7YK5v6dCeB0fxJ-{',
            'RESOURCE_INSTANCE_ID': 'crn:v1:bluemix:public:cloud-object-storage:global:a/a133b10269884cd3bd70e68f7ab0c5d5:8085c05a-5cab-4b6a-bd17-8ef09b3a87fb::',
            'ACCESS_KEY': 'e734567eb7e54165a7c6fb017a0a02f3'
        }
```

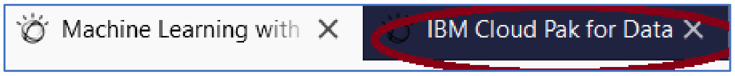
Step 5: Generate API Key to interface to Watson Machine Learning Service

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.

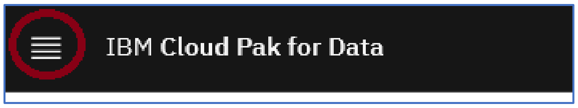
1. Right-click on **IBM Cloud Pak for Data**, and the click on **Open Link in New Tab**.

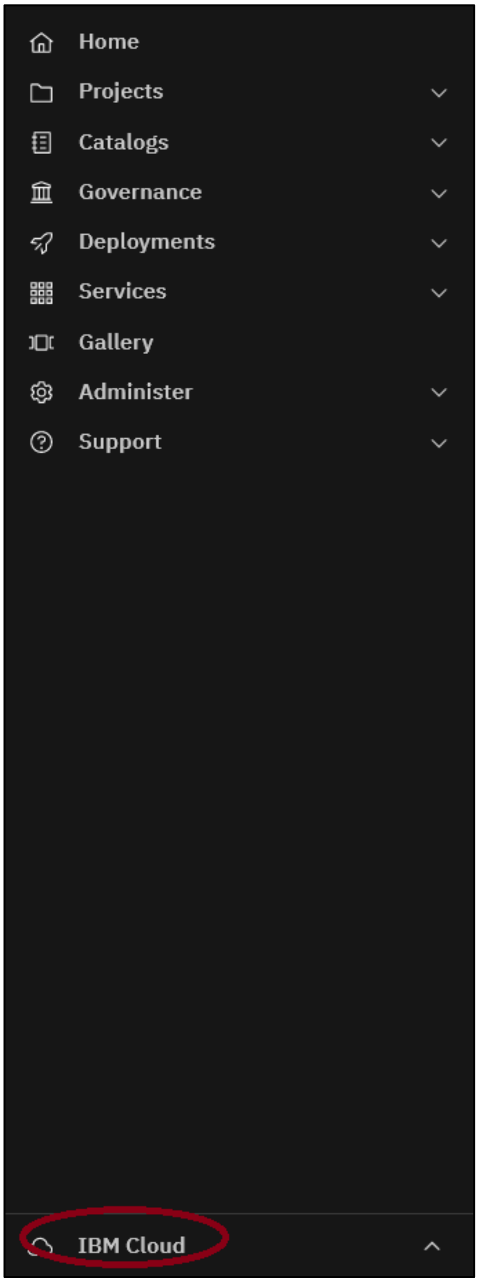

IBM Cloud Pak for Data
Search
Projects / Watson Studio La
File
Edit
View
Insert
Markdown
Open Link in New Tab
Open Link in New Window
Open Link in New Private Window
Bookmark This Link
Save Link As...
Save Link to Pocket
Copy Link Location
Search Google for "IBM Cloud Pak f..."
Send Link to Device
Inspect Accessibility Properties
Inspect Element (Q)
Machin
ML
In this notebook, we will explore machine learning using Spark
to create and tune machine learning pipelines. Spark ML Pipelin
workflow. We will utilize Spark ML's feature transformers to con

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

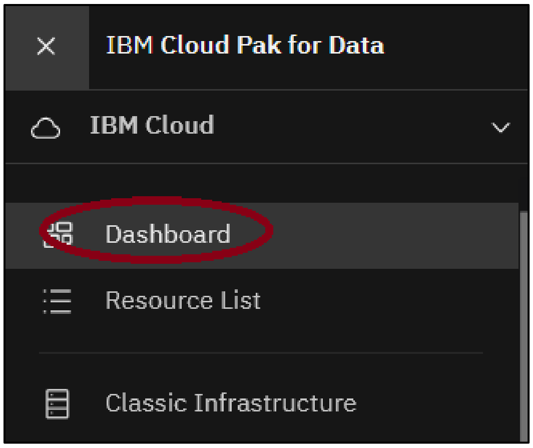


3. Click on the hamburger icon ☰, and then scroll down to the bottom to click on **IBM Cloud**.

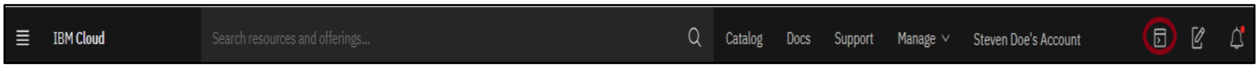




4. Click on **Dashboard**



5. Click on the IBM Command Shell icon.



6. 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**.

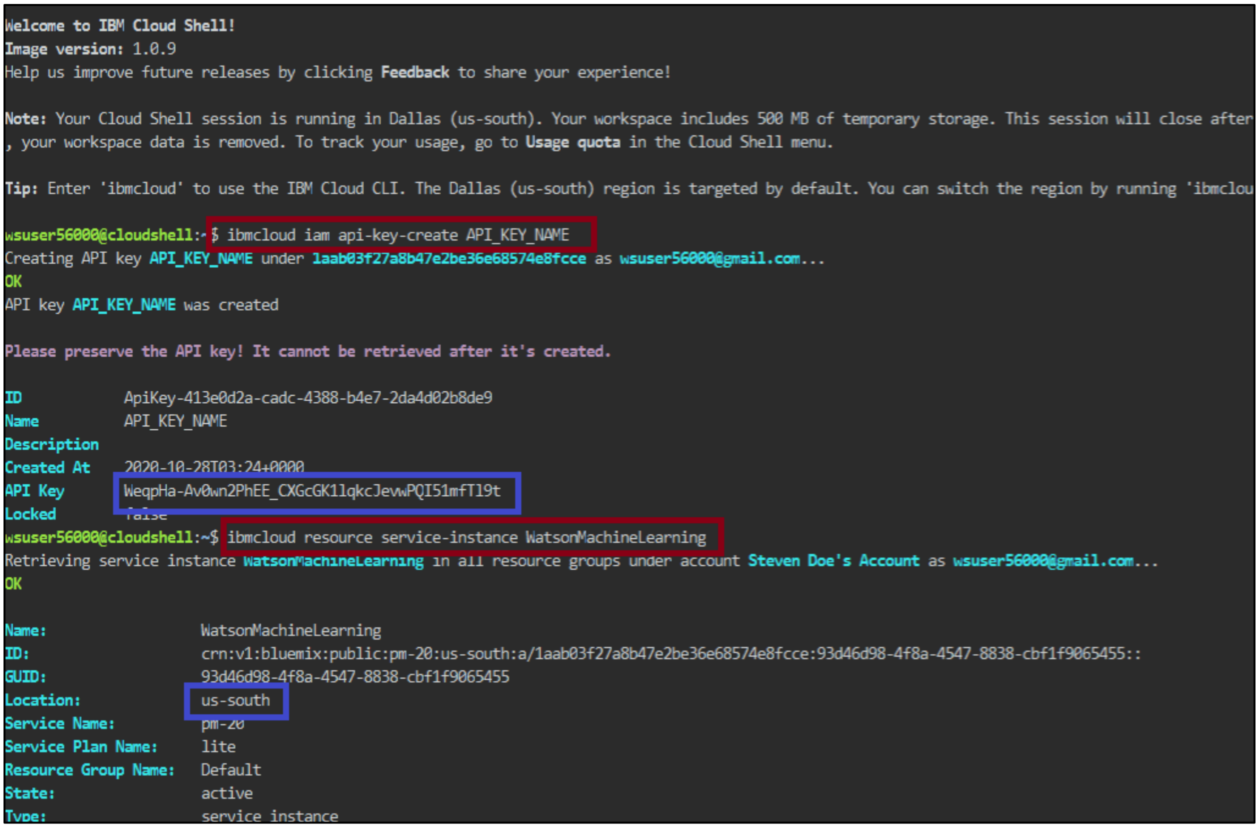
The newly created api key is shown highlighted in **blue** below. This key is unique to my account. Your key will be unique to your account.

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**.

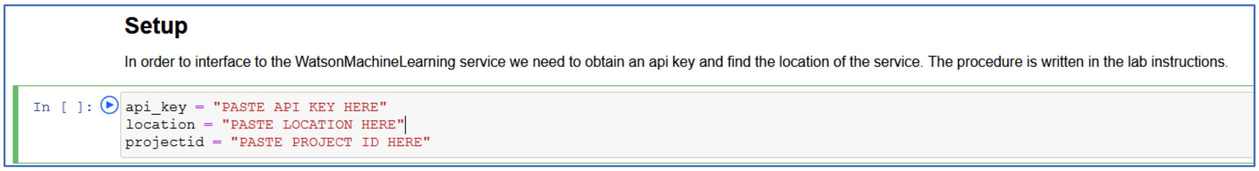
The location of the WatsonMachineLearning instance is shown highlighted in **blue** below. My instance location is shown to be in us-south.



```
Welcome to IBM Cloud Shell!
Image version: 1.0.9
Help us improve future releases by clicking Feedback to share your experience!

Note: Your Cloud Shell session is running in Dallas (us-south). Your workspace includes 500 MB of temporary storage. This session will close after
, your workspace data is removed. To track your usage, go to Usage quota in the Cloud Shell menu.

Tip: Enter 'ibmcloud' to use the IBM Cloud CLI. The Dallas (us-south) region is targeted by default. You can switch the region by running 'ibmclou

wsuser56000@cloudshell:~$ ibmcloud iam api-key-create API_KEY_NAME
Creating API key API_KEY_NAME under 1aab03f27a8b47e2be36e68574e8fcce as wsuser56000@gmail.com...
OK
API key API_KEY_NAME was created

Please preserve the API key! It cannot be retrieved after it's created.

ID            ApiKey-413e0d2a-cadc-4388-b4e7-2da4d02b8de9
Name          API_KEY_NAME
Description
Created At    2020-10-28T03:24+0000
API Key       WeqpHa-Av0wn2PhEE_CXGcGK1lqkcJevwPQI51mfT19t
Locked        false
wsuser56000@cloudshell:~$ ibmcloud resource service-instance WatsonMachineLearning
Retrieving service instance WatsonMachineLearning in all resource groups under account Steven Doe's Account as wsuser56000@gmail.com...
OK

Name:                   WatsonMachineLearning
ID:                     crn:v1:bluemix:public:pm-20:us-south:a/1aab03f27a8b47e2be36e68574e8fcce:93d46d98-4f8a-4547-8838-cbf1f9065455::
GUID:                   93d46d98-4f8a-4547-8838-cbf1f9065455
Location:               us-south
Service Name:           pm-20
Service Plan Name:      lite
Resource Group Name:    Default
State:                  active
Type:                   service instance
```

7. 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:



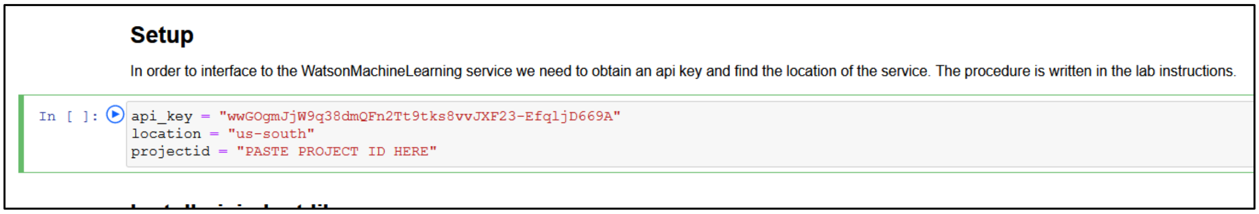
Setup

In order to interface to the WatsonMachineLearning service we need to obtain an api key and find the location of the service. The procedure is written in the lab instructions.

```
In [ ]: api_key = "PASTE API KEY HERE"
        location = "PASTE LOCATION HERE"
        projectid = "PASTE PROJECT ID HERE"
```

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

To:



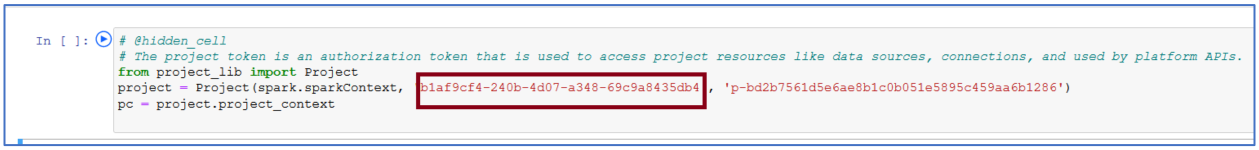
Setup

In order to interface to the WatsonMachineLearning service we need to obtain an api key and find the location of the service. The procedure is written in the lab instructions.

```
In [ ]: api_key = "wwGOgmJjW9q38dmQFn2Tt9tks8vvJXF23-EfqljD669A"
        location = "us-south"
        projectid = "PASTE PROJECT ID HERE"
```

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

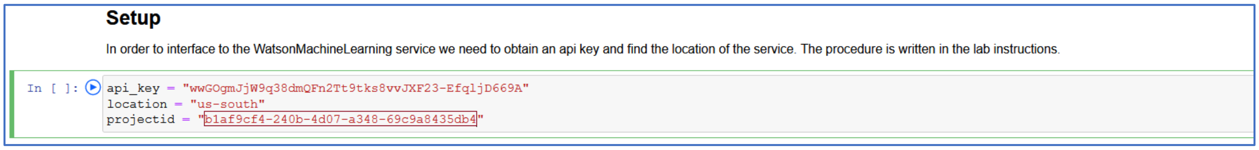
Copy:



```
In [ ]: # @hidden_cell
        # The project token is an authorization token that is used to access project resources like data sources, connections, and used by platform APIs.
        from project_lib import Project
        project = Project(spark.sparkContext, 'b1af9cf4-240b-4d07-a348-69c9a8435db4', 'p-bd2b7561d5e6ae8b1c0b051e5895c459aa6b1286')
        pc = project.project_context
```

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

To:



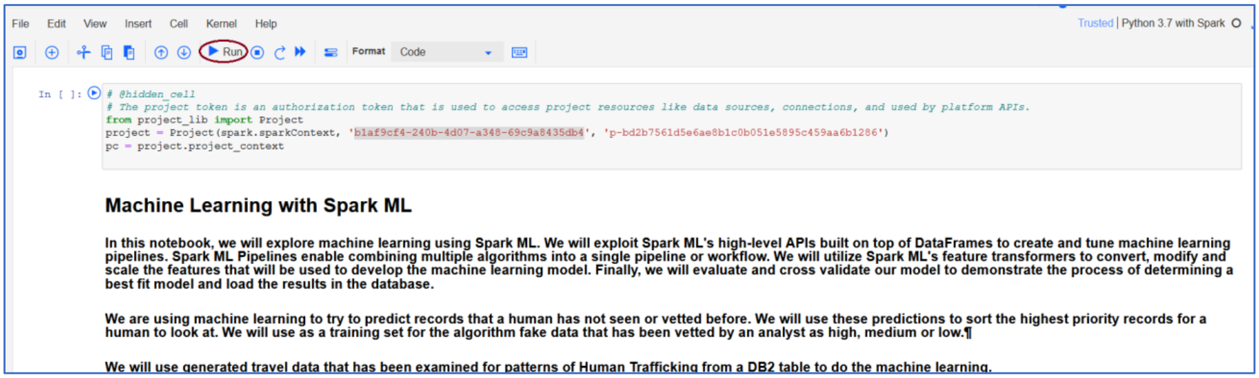
Setup

In order to interface to the WatsonMachineLearning service we need to obtain an api key and find the location of the service. The procedure is written in the lab instructions.

```
In [ ]: api_key = "wwGOgmJjW9q38dmQFn2Tt9tks8vvJXF23-EfqljD669A"
        location = "us-south"
        projectid = "b1af9cf4-240b-4d07-a348-69c9a8435db4"
```

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.**



```
In [ ]: # @hidden_cell
        # The project token is an authorization token that is used to access project resources like data sources, connections, and used by platform APIs.
        from project_lib import Project
        project = Project(spark.sparkContext, 'b1af9cf4-240b-4d07-a348-69c9a8435db4', 'p-bd2b7561d5e6ae8b1c0b051e5895c459aa6b1286')
        pc = project.project_context
```

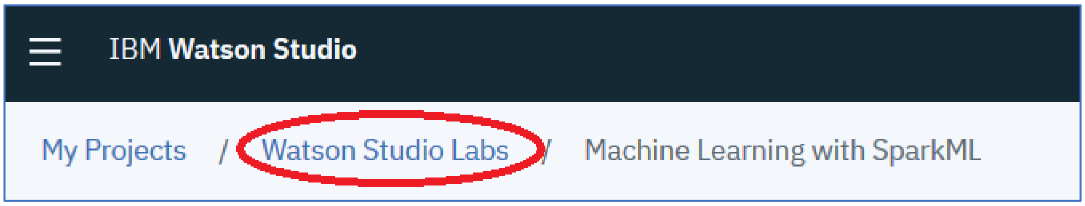
Machine Learning with Spark ML

In this notebook, 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. Spark ML Pipelines enable combining multiple algorithms into a single pipeline or workflow. 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 and load the results in the database.

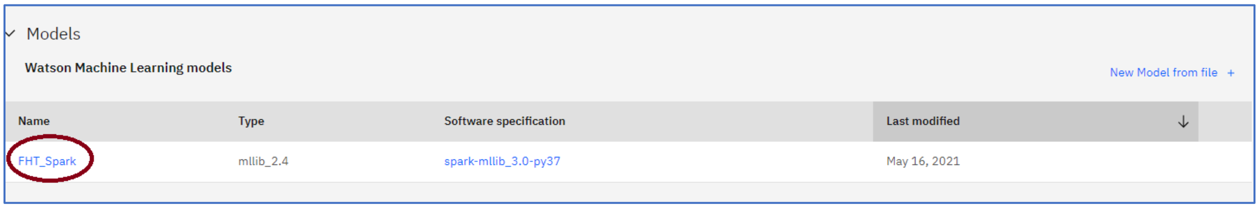
We are using machine learning to try to predict records that a human has not seen or vetted before. We will use these predictions to sort the highest priority records for a human to look at. We will use as a training set for the algorithm fake data that has been vetted by an analyst as high, medium or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

We will use generated travel data that has been examined for patterns of Human Trafficking from a DB2 table to do the machine learning.

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


IBM Watson Studio
My Projects / Watson Studio Labs / Machine Learning with SparkML

3. 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.



Models

Watson Machine Learning models

Name	Type	Software specification	Last modified
FHT_Spark	mllib_2.4	spark-mllib_3.0-py37	May 16, 2021

You have completed Lab-5!

- ✓ Created 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.
- ✓ Saved the model to the model repository.