

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.

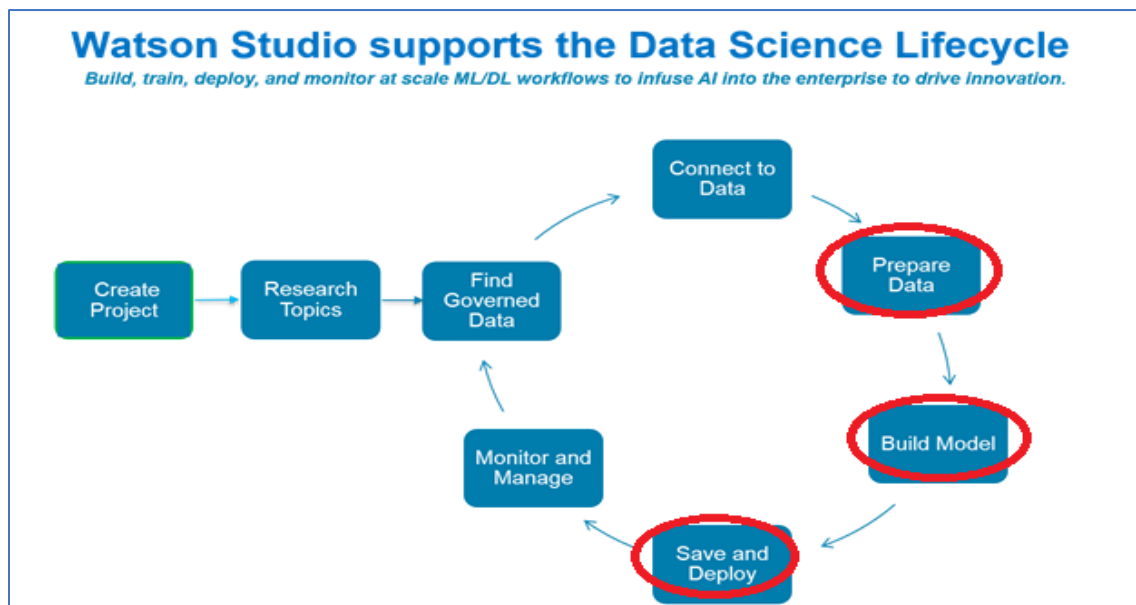


Figure 1- End to End Flow

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 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.

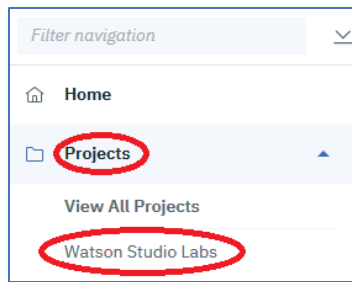
Other columns in the dataset are similarly very granular and could also be categorized for modeling purposes. This lab does not include steps to accomplish this, but it would be similar to what was done for the occupation column.

Lab Steps

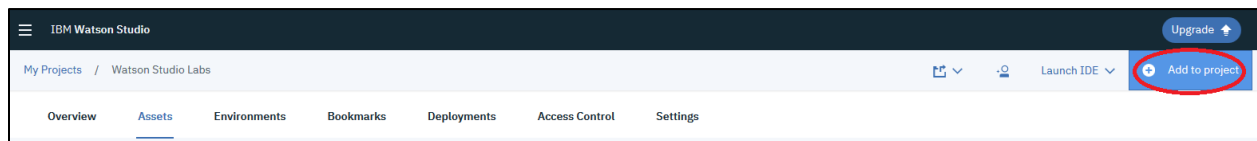
Step 1 - Create a Jupyter Notebook

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

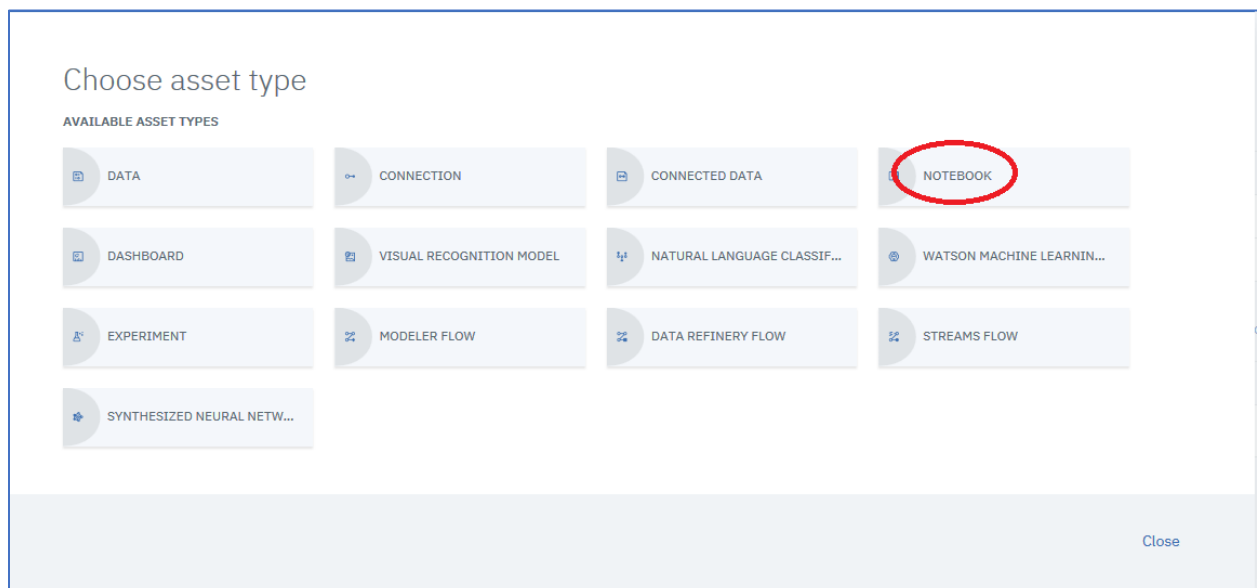




2. We are now going to create a notebook in our project. This notebook will be created from a url that points to the Machine Learning with SparkML notebook in the github repository. Click the **Add to project** link.



3. Click on **NOTEBOOK**



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

Select the Runtime. You will need to change the default selection. MAKE SURE TO SELECT Default Spark Python 3.6 XS (Driver with 1vCPU ...

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

https://github.com/bleonardb3/DS_POT_09-05/blob/master/Lab-5/Machine%20Learning%20with%20SparkML.ipynb

Click **Create Notebook**.

My Projects / Watson Studio Labs / Add Notebook

New notebook

Blank From file **From URL**

Name
Machine Learning with SparkML 11 characters remaining

Description (optional)
Type your Description here 500 characters remaining

Select runtime
Default Spark Python 3.6 XS (Driver with 1 vCPU and 4 GB RAM, 2 executors with 1 vCPU and 4 GB RAM)
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.

Notebook URL
https://github.com/bleonardb3/DS_POT_09-05/blob/master/Lab-5/Machine%20Learning%20with%20SparkML.ipynb

Cancel **Create Notebook**

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

My Projects / Watson Studio Labs / Machine Learning with SparkML

File Edit View Insert Cell Kernel Help

Kernel ready Trusted **Python 3.6 with Spark**

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.

6. 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.

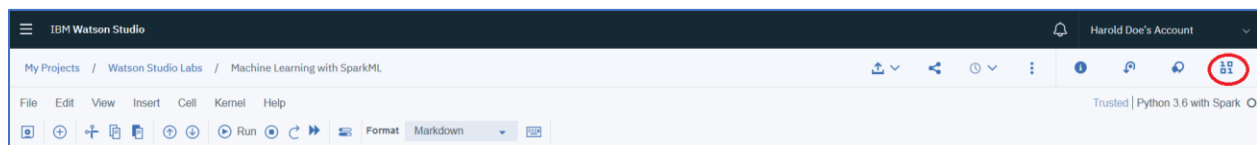
Step 2: Insert Generated Code

1. Before executing the cells in the notebook, we are going to use the IBM value-add code generator to insert code in 4 code cells. Scroll down in the notebook until you see **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).

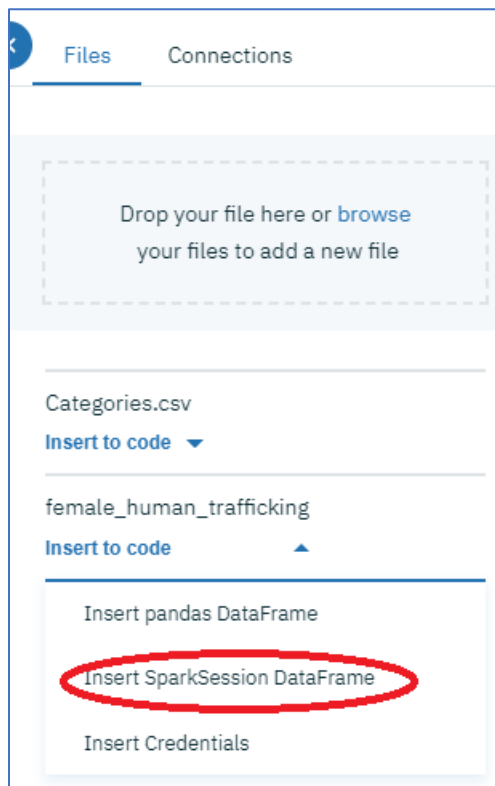
```
Read Data Asset - female_human_trafficking - See Lab Instructions

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.
```

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



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



4. Locate the variable `df_data_n` (n is a number). This is a generated variable. We need to change this variable name to **trafficking_df**.

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

#@hidden_cell
# The following code is used to access your data and contains your credentials.
# You might want to remove those credentials before you share your notebook.

properties_b13f4a659ddb4b3aa53ffd2f142e4912 = {
    'jdbcurl': 'jdbc:db2://dashdb-entry-yp-dal09-08.services.dal1 Bluemix.net:50000/BLUDB',
    'user': 'dash100316',
    'password': 'GvEI{uLxgr4n'
}

data_df_1 = spark.read.jdbc(properties_b13f4a659ddb4b3aa53ffd2f142e4912['jdbcurl'], table='DASH100316.FEMALE_HUMAN_TRAFFICKING', properties=properties_b13f4a659ddb4b3aa53ffd2f142e4912)
data_df_1.head()
```

Change to.

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

#@hidden_cell
# The following code is used to access your data and contains your credentials.
# You might want to remove those credentials before you share your notebook.

properties_b13f4a659ddb4b3aa53ffd2f142e4912 = {
    'jdbcurl': 'jdbc:db2://dashdb-entry-yp-dal09-08.services.dal1 Bluemix.net:50000/BLUDB',
    'user': 'dash100316',
    'password': 'GvEI{uLxgr4n'
}

trafficking_df = spark.read.jdbc(properties_b13f4a659ddb4b3aa53ffd2f142e4912['jdbcurl'], table='DASH100316.FEMALE_HUMAN_TRAFFICKING', properties=properties_b13f4a659ddb4b3aa53ffd2f142e4912)
trafficking_df.head()
```


5. Scroll down to **Read Data Asset – Occupations – See Lab Instructions**. Click 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 **Insert to code**  down arrow underneath **Occupation**. Click on **Insert Credentials**.

Categories

Insert to code ▼

Occupation

Insert to code ▲

Insert Credentials

Insert to code ▼

- 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'
}
```

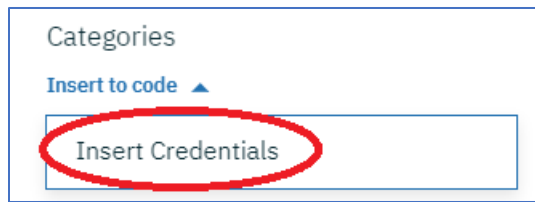
Change to:

```
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'
}
```

- 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
```

- Click on **Insert to code** down arrow underneath **Categories**. Click on **Insert Credentials**.



10. 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_2 {
  'bucket': 'watsonstudiolabs-datacatalog-b13amkzfb',
  'FILE': 'data_asset',
  'URL': 'https://s3.us-south.objectstorage.softlayer.net',
  'SECRET_KEY': '8491e438088d1b0e217eacc792850ee38c1e72d89537be7d',
  'API_KEY': 't1DsT16tH6P1XT6nRnvMCZxRmobwI7YK5v6dCeB0fxJ-{'',
  'RESOURCE_INSTANCE_ID': 'crn:v1:bluemix:public:cloud-object-storage:global:a/a133b10269884cd3bd70e68f7ab0c5d5:8085c05a-5cab-4b6a-bd17-8ef09b3a87fb::',
  'ACCESS_KEY': 'e734567eb7e54165a7c6fb017a0a02f3'
}
```

Change to:

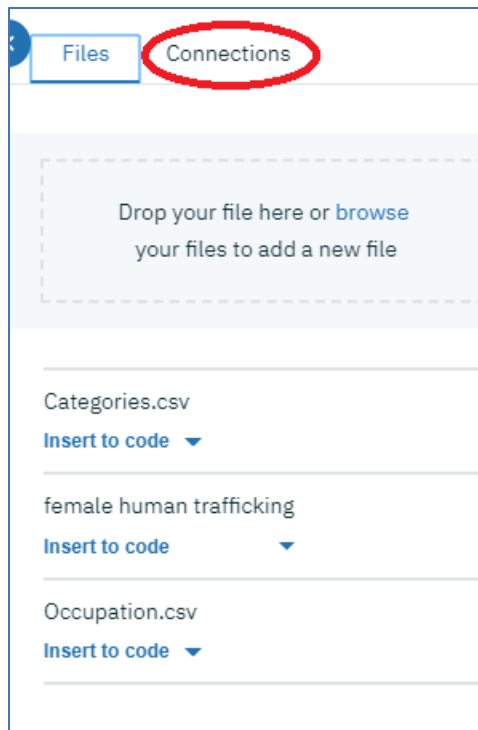
```
In [ ]: # Insert Categories credentials in this cell after the comments
# make CERTAIN that if credentials gets generated as credentials_1 or credentials_2 or credentials_n where n is a number to rename as credentials
# Put cursor on the next line to Insert to code
#@hidden_cell
# The following code 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': 't1DsT16tH6P1XT6nRnvMCZxRmobwI7YK5v6dCeB0fxJ-{'',
  'RESOURCE_INSTANCE_ID': 'crn:v1:bluemix:public:cloud-object-storage:global:a/a133b10269884cd3bd70e68f7ab0c5d5:8085c05a-5cab-4b6a-bd17-8ef09b3a87fb::',
  'ACCESS_KEY': 'e734567eb7e54165a7c6fb017a0a02f3'
}
```

11. Scroll down further towards the middle of the notebook until you see **Insert the database credentials – see Lab Instructions**. Click cursor underneath the commented lines in the code cell.

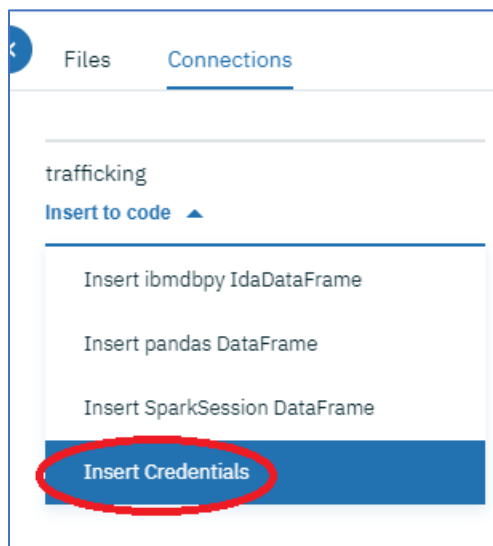
```
Insert the database credentials - see Lab Instructions

In [ ]: # Insert database connection credentials below
# Make sure the name that is used is credentials. If credentials_1 is shown, please change to credentials.
```

12. Click on **Connections**.



13. Click on [Insert to code](#) ▼ down arrow underneath **trafficking**. Click on **Insert Credentials**.



14. 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 the database credentials - see Lab Instructions

```
In [ ]: # Insert database connection credentials below
# Make sure the name that is used is credentials. If credentials_1 is shown, please change to credentials.
# @hidden_cell
# The following code contains the credentials for a connection in your Project.
# You might want to remove those credentials before you share your notebook.
credentials = {
    'username': 'dash100316',
    'password': '""GvEI{uLxgr4r""',
    'sg_service_url': 'https://sgmanager.ng.bluemix.net',
    'database': 'BLUDB',
    'host': 'dashdb-entry-yp-dal09-08.services.dal.bluemix.net',
    'port': '50000',
    'url': 'https://undefined'
}
```

Change to:

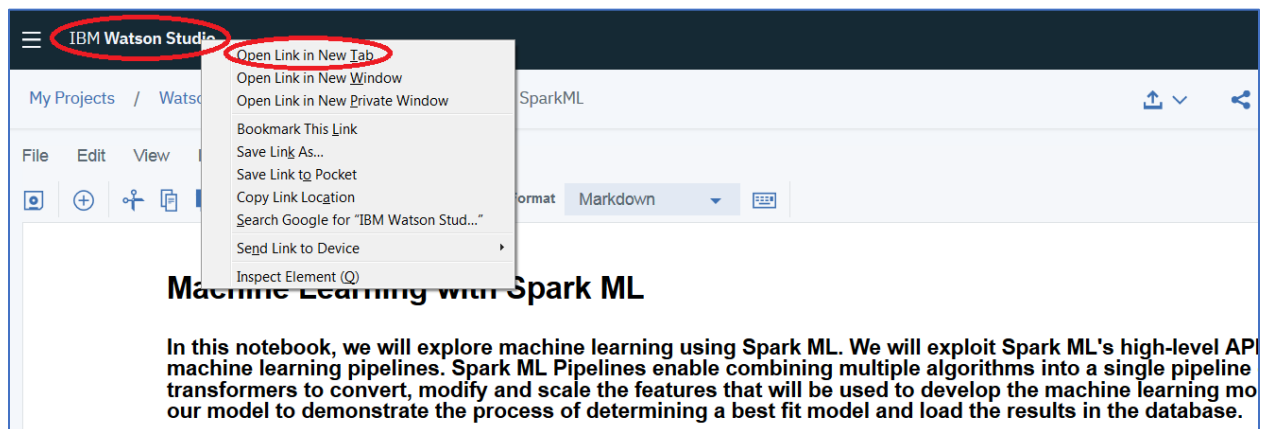
Insert the database credentials - see Lab Instructions

```
In [ ]: # Insert database connection credentials below
# Make sure the name that is used is credentials. If credentials_1 is shown, please change to credentials.
# @hidden_cell
# The following code contains the credentials for a connection in your Project.
# You might want to remove those credentials before you share your notebook.
credentials = {
    'username': 'dash100316',
    'password': '""GvEI{uLxgr4r""',
    'sg_service_url': 'https://sgmanager.ng.bluemix.net',
    'database': 'BLUDB',
    'host': 'dashdb-entry-yp-dal09-08.services.dal.bluemix.net',
    'port': '50000',
    'url': 'https://undefined'
}
```

Step 3: Copy Watson Machine Learning Credentials

In the notebook we will save our model to the Watson Machine Learning service model repository. We need to get the credentials of the Watson Machine Learning service and will copy and paste these credentials into the appropriate notebook cell.

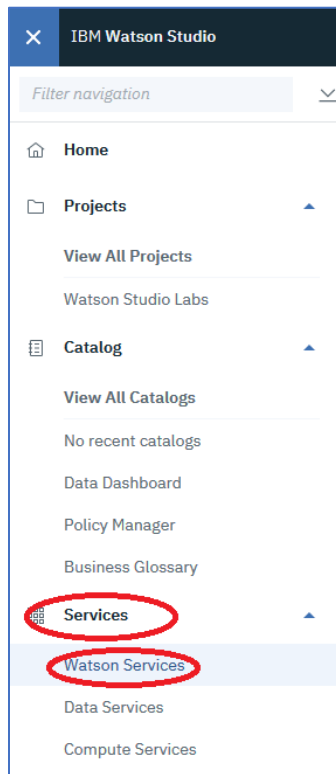
1. Right-click on **IBM Watson Studio**, and then click on **Open Link in New Tab**.




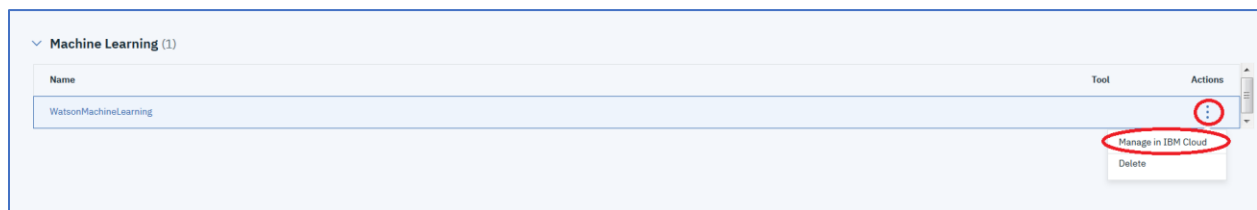
2. Click on the new **Watson Studio** browser tab.



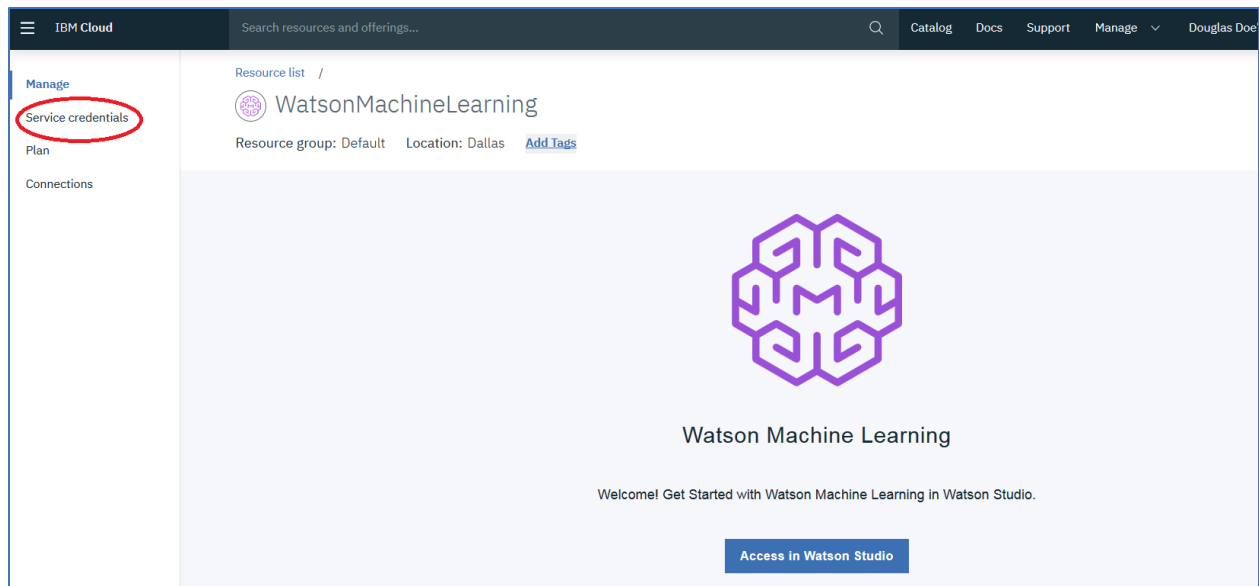
3. Click on the hamburger icon , and then **Services**, and **Watson Services**.



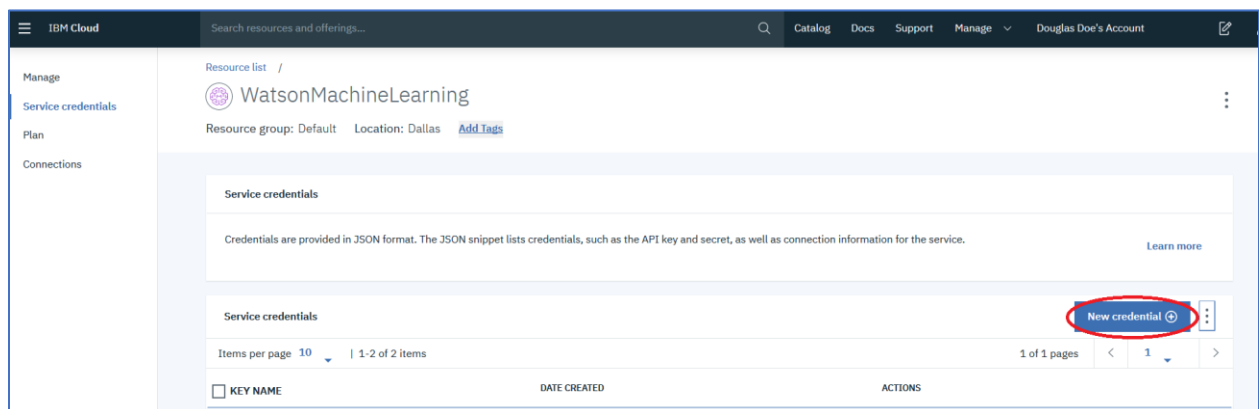
4. Hover over WatsonMachineLearning, click on the vertical ellipse on the right. , and click on **Manage in IBM Cloud**.



5. A new browser tab will be created titled **Service Details – IBM Cloud**. This browser tab will be interfacing with the IBM Cloud user interface. Click on **Service credentials**.



6. Click on **New Credentials+**.



7. Click on **Add**.

×

Add new credential

Name:

Service credentials-1

Role:

Writer

Select Service ID (Optional)

Select Service ID...

Add Inline Configuration Parameters (Optional):

Cancel

Add

8. Click on ▼ next to View Credentials in the Service Credentials-1 row.

Service credentials


Credentials are provided in JSON format. The JSON snippet lists credentials, such as the API key and secret, as well as connection information for the service. [Learn more](#)

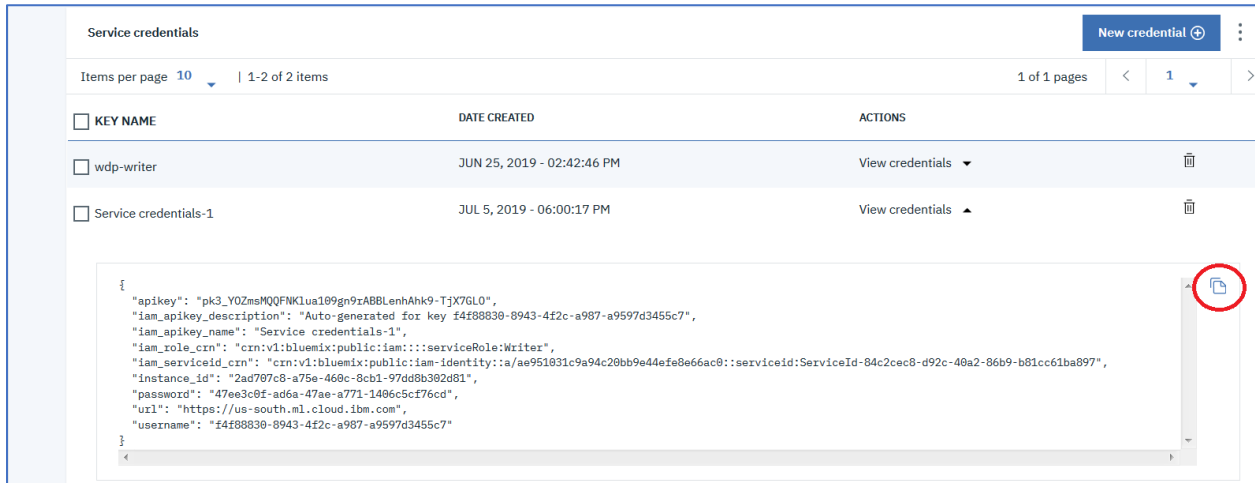
Service credentials

New credential

Items per page 10 | 1-2 of 2 items 1 of 1 pages < 1 >





<input type="checkbox"/> KEY NAME	DATE CREATED	ACTIONS
<input type="checkbox"/> wdp-writer	JUN 25, 2019 - 02:42:46 PM	View credentials <div></div> <div></div>
<input type="checkbox"/> Service credentials-1	JUL 5, 2019 - 06:00:17 PM	View credentials <div></div> <div></div>

9. Click on the copy icon 



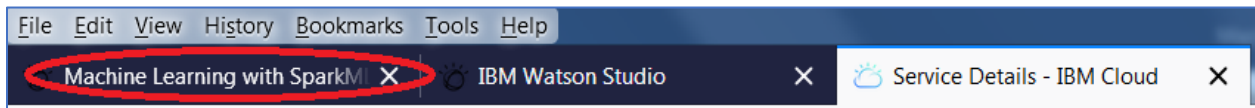
Service credentials

Items per page 10 | 1-2 of 2 items 1 of 1 pages < 1 >

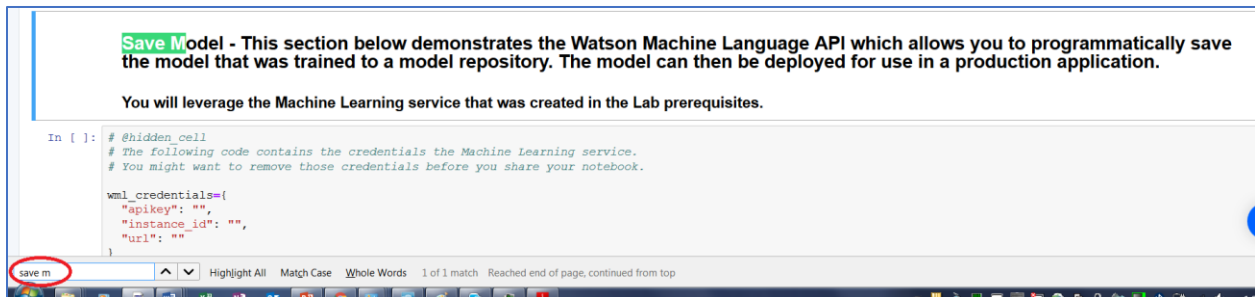
KEY NAME	DATE CREATED	ACTIONS
<input type="checkbox"/> wdp-writer	JUN 25, 2019 - 02:42:46 PM	View credentials  
<input type="checkbox"/> Service credentials-1	JUL 5, 2019 - 06:00:17 PM	View credentials  

```
{
  "apikey": "pk3_Y0ZmsMQQFNK1ua189gn9rABBLenhAhk9-TjX7GL0",
  "iam_apikey_description": "Auto-generated for key f4f88830-8943-4f2c-a987-a9597d3455c7",
  "iam_apikey_name": "Service credentials-1",
  "iam_role_crn": "crn:v1:bluemix:public:iam::::serviceRole:Writer",
  "iam_serviceid_crn": "crn:v1:bluemix:public:iam-identity::a/ae951831c9a94c28bb9e44efe8e66ac8::serviceid:ServiceId-84c2cec8-d92c-40a2-86b9-b81cc61ba897",
  "instance_id": "2ad707c8-a75e-460c-8cb1-97dd8b302d81",
  "password": "47ee3c0f-ad6a-47ae-a771-1486c5cf76cd",
  "url": "https://us-south.ml.cloud.ibm.com",
  "username": "f4f88830-8943-4f2c-a987-a9597d3455c7"
}
```

10. Click on the Machine Learning browser tab to go back to the notebook. Should be two browser tabs to the left of the **Service Details – IBM Cloud**.



11. Type Ctrl-F and type in **save m** to find the notebook cell to paste in the credentials.



Save Model - This section below demonstrates the Watson Machine Language API which allows you to programmatically save the model that was trained to a model repository. The model can then be deployed for use in a production application.

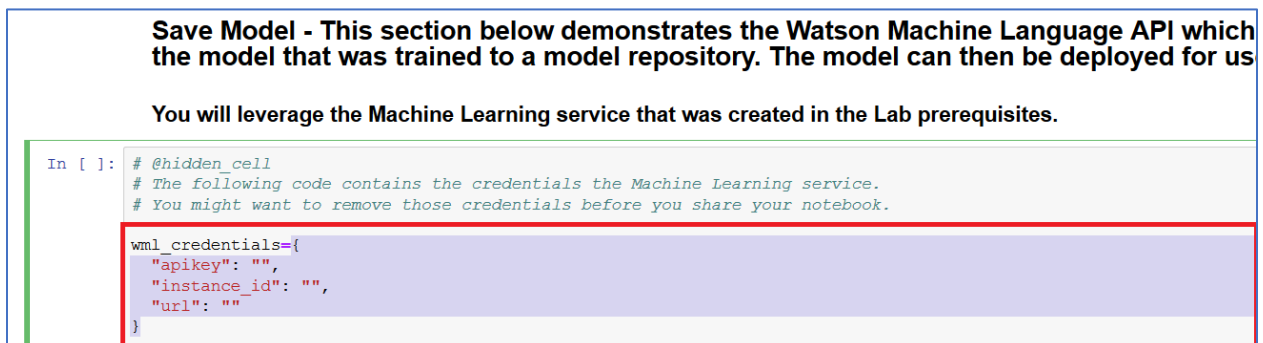
You will leverage the Machine Learning service that was created in the Lab prerequisites.

```
In [ ]: # @hidden_cell
# The following code contains the credentials the Machine Learning service.
# You might want to remove those credentials before you share your notebook.

wml_credentials={
  "apikey": "",
  "instance_id": "",
  "url": ""
}
```

save m

12. Highlight the text from the starting curly brace { to the ending curly brace }.



Save Model - This section below demonstrates the Watson Machine Language API which the model that was trained to a model repository. The model can then be deployed for us

You will leverage the Machine Learning service that was created in the Lab prerequisites.

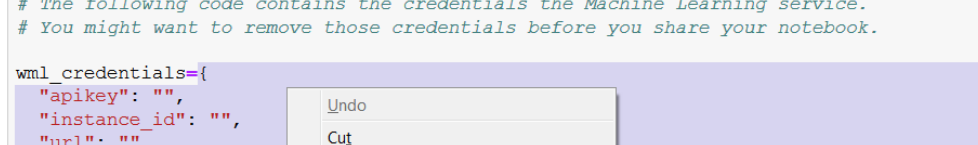
```
In [ ]: # @hidden_cell
# The following code contains the credentials the Machine Learning service.
# You might want to remove those credentials before you share your notebook.

wml_credentials={
  "apikey": "",
  "instance_id": "",
  "url": ""
}
```

13. Right-click on the highlighted text and click **Paste**.

Save Model - This section below demonstrates the Watson Machine Language the model that was trained to a model repository. The model can then be dep

You will leverage the Machine Learning service that was created in the Lab prerequisites.



The screenshot shows a Jupyter Notebook interface with a code cell. The code defines a dictionary for Watson Machine Learning credentials. A context menu is open over the code, with the 'Paste' option highlighted. The code is as follows:

```
In [ ]: # @hidden_cell
# The following code contains the credentials the Machine Learning service.
# You might want to remove those credentials before you share your notebook.

wml_credentials={
    "apikey": "",
    "instance_id": "",
    "url": ""
}
```

The context menu options visible are: Undo, Cut, Copy, Paste (highlighted), Delete, Select All, Search Google for "{ 'apikey': '', '...',", This Frame, View Selection Source, Check Spelling, and Inspect Element (Q).

14. The credentials should appear similar to below with different values.

Save Model - This section below demonstrates the Watson Machine Language API which allows you to programmatically save the model that was trained to a model repository. The model can then be deployed for use in a production application.

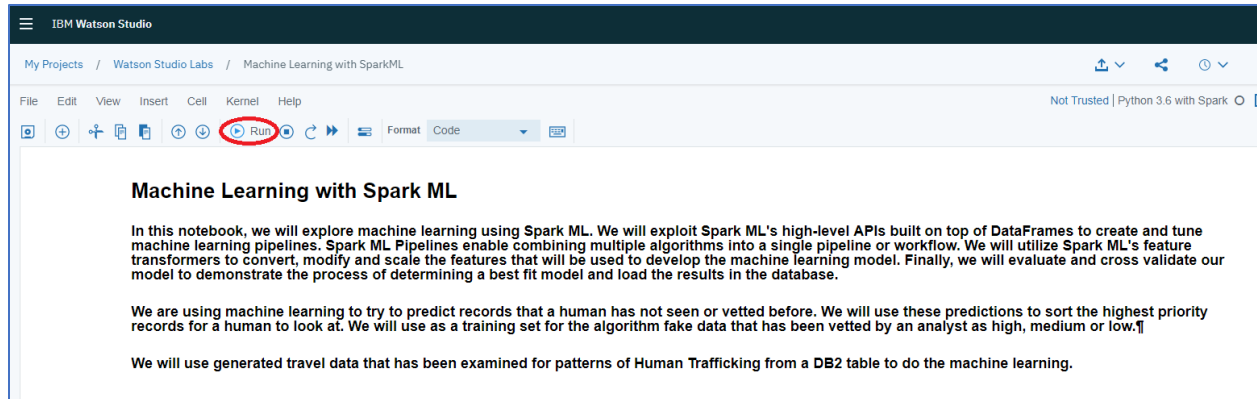
You will leverage the Machine Learning service that was created in the Lab prerequisites.

```
In [ ]: # @hidden_cell
# The following code contains the credentials the Machine Learning service.
# You might want to remove those credentials before you share your notebook.

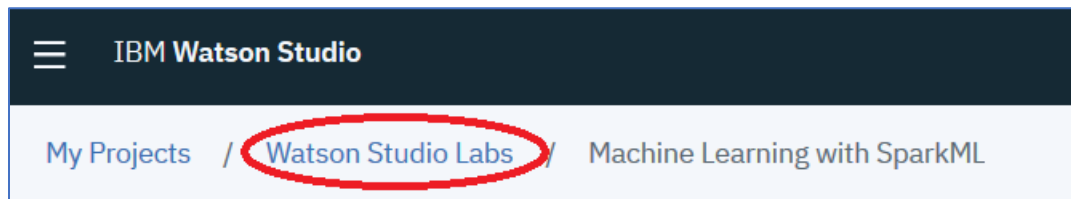
wml_credentials={
    "apikey": "r5XhQcEliuBXRszGKy5zTm7M0dMPp8ZISLINVJHdM2oY",
    "iam_apikey_description": "Auto-generated for key 4df948a5-a679-4dba-aaf0-ba380ae21396",
    "iam_apikey_name": "Service credentials-1",
    "iam_role_arn": "arn:aws:iam::public:role::serviceRole:Writer",
    "iam_serviceid": "arn:aws:iam::public:serviceid::a/a133b10269884cd3bd70e6f7ab0c5d5::serviceid:ServiceId-b6d0836d-df64-44b1-957c-9f51f7363e2d",
    "instance_id": "083ff2e8-0201-41c7-8796-e3c6bd77010",
    "url": "https://us-south.ml.cloud.ibm.com"
}
```

Step 4: 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.**



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



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.

You have completed Lab-5!

- ✓ 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.

