

# **Watson Machine Learning Overview**

This lab will introduce the Watson Machine Learning capability using the Titanic dataset. The lab will consist of the following steps:

1. Setting up the environment
2. Adding a data asset to the DSX Labs project
3. Creating a Model to predict whether a person would survive
4. Deploying and Testing the Model
5. Creating a simple web front-end

## **Step 1: Setting up your environment**

To use IBM Watson Machine Learning you must have the following service instances in your Bluemix dashboard:

- Watson Machine Learning
- Object Storage
- Apache Spark

The Object Storage and Apache Spark service instances should already exist having been created when your DSX account was provisioned. We now need to provision a Machine Learning Service.

### **Step 1.1: Creating a Machine Learning Instance**

To create a Machine Learning service instance, you must perform the following steps:

1. Go to the **Settings** tab in your project

The screenshot shows the 'Project Information' section of a project named 'LMCO'. The 'Settings' tab is highlighted with a red arrow pointing to it. The 'Project Name' field contains 'LMCO'. The 'Description' field has a placeholder 'Project Description' and a character count of 3000. A 'Save' button is visible at the bottom right.

2. Scroll down to **Associated Services** select **add associated service** and select **Machine Learning**

The screenshot shows the 'Associated Services' list. It includes a table with columns: NAME, SERVICE TYPE, and PLAN. One entry is 'DSX-Spark' (Spark, Personal). A dropdown menu is open, showing options: 'Amazon EMR Spark', 'IBM Analytics Engine', 'Machine Learning', and 'Spark'. A blue arrow points from the 'Machine Learning' option in the dropdown to the main list.

3. Select the **Free** instance and create

The screenshot shows the 'Add Machine Learning Service' page. The 'New' tab is selected. Under 'Machine Learning', there is a description of the service. In the 'Features' section, it lists 'SPSS analytics platform features' and 'Coming Soon! Integration with Data Science Experience'. In the 'Pricing Plan' section, it says 'Monthly Process shown above reflect the: United States'. Below this, there is a table with columns: Plan, Features, and Pricing. The 'Plan' column has a 'Free' option selected, indicated by a red circle and a blue arrow pointing to it. The 'Features' column lists 'Service instance (5 models per instance)', '5,000 predictions', and '5 compute hours'. The 'Pricing' column shows 'Free'.

4. Give a name to your new Machine Learning instance and **Confirm**

## Confirm Purchase

Organization: dsxspare3@gmail.com

Plan  
**Free** ▾

Space  
**DataSciX** ▾

Service Name  
**WML Instance** ▾

Cancel **Confirm**



## Step 2: Adding a Data Asset to the project

1. Download the Titanic data file from

[https://github.com/jpatter/ML-POT/blob/master/Lab-2/data/titanic\\_cleaned.csv](https://github.com/jpatter/ML-POT/blob/master/Lab-2/data/titanic_cleaned.csv)

The data in this file has already been prepared and it ready to be input into the Modeling step. Select the link above.

2. Right click on Raw, and click on Save link as ....

A screenshot of a GitHub repository page for 'jpatter / ML-POT'. The repository has 1 star, 1 watch, and 0 forks. The 'Code' tab is selected. Below it, the file 'titanic\_cleaned.csv' is shown. The file has 1045 lines, 1044 sloc, and 75.6 KB. It was added by 'jpatter' 6 hours ago. A blue arrow points to the 'Raw' button in the file's metadata section.

1	pclass	survived	name	sex	sibsp	parch	ticket	fare	embarked
2	1	1	Allison, Miss. Elisabeth Walton	female	0	0	24160	211.337500	S
3	1	1	Allison, Master. Hudson Trevor	male	1	2	113781	151.550000	S
4	1	0	Allison, Miss. Helen Loraine	female	1	2	113781	151.550000	S
5	1	0	Allison, Mr. Hudson Joshua Creighton	male	1	2	113781	151.550000	S
6	1	n	Allison, Mr. Hudson Joshua Creighton	female	1	2	113781	151.550000	S

3. Go back to your DSX project. Click on **add data assets** or the  icon.

The screenshot shows the IBM Data Science Experience web interface. The top navigation bar includes 'File', 'Edit', 'View', 'History', 'Bookmarks', 'Tools', and 'Help'. Below the bar, there are several tabs: 'Overview - IBM Machi...', 'Tutorial: Build a logist...', 'Mail', 'Dashboard - IBM Blue...', 'IBM Data Science Exp...', 'Copy file from github ...', 'Proof-of-Technology/...', and 'Proof-of-Technology/...'. The main content area is titled 'My Projects > DSX Labs'. It features sections for 'Overview', 'Analytics Assets', 'Data Assets', 'Bookmarks', 'Deployments', 'Collaborators', and 'Settings'. The 'Data Assets' section is currently selected, displaying a table with columns: NAME, SHARED, STATUS, LANGUAGE, LAST EDITOR, LAST MODIFIED, and ACTIONS. Two entries are listed: 'Machine Learning with Spark - Joel' (Python 3.5, Bernard Beekman, 11 Jul 2017) and 'Connect and Interact with DashDB' (Python 3.5, Bernard Beekman, 10 Jul 2017). To the right of the table is a button labeled '+ add data assets' with a plus sign icon. A blue arrow points to this button.

- Click on browse and then go to the folder where the titanic\_cleaned.csv is stored. Select titanic\_cleaned.csv and then click Open.

The screenshot shows the same IBM Data Science Experience interface as the previous one, but with a different set of data assets. The 'Data Assets' section is selected, showing a table with columns: NAME, TYPE, SERVICE, LAST MODIFIED, and ACTIONS. Three entries are listed: 'BikeShare.csv' (CSV, Object Storage (Swift API)), 'Female-human-trafficking.csv' (CSV, Object Storage (Swift API)), and 'myDashDB' (Connection, dashDB, 5 Jul 2017). To the right of the table is a sidebar with tabs for 'Files' and 'Connections'. The 'Files' tab is active, showing a large dashed box with the text 'Drop file here or browse your files to add a new file'. Below this is a 'Find in Storage' search bar and a list of checked files: 'Female-human-trafficking.csv' and 'BikeShare.csv'. A blue arrow points to the 'Drop file here or browse your files to add a new file' box.

## Step 3: Create a Model to predict survival

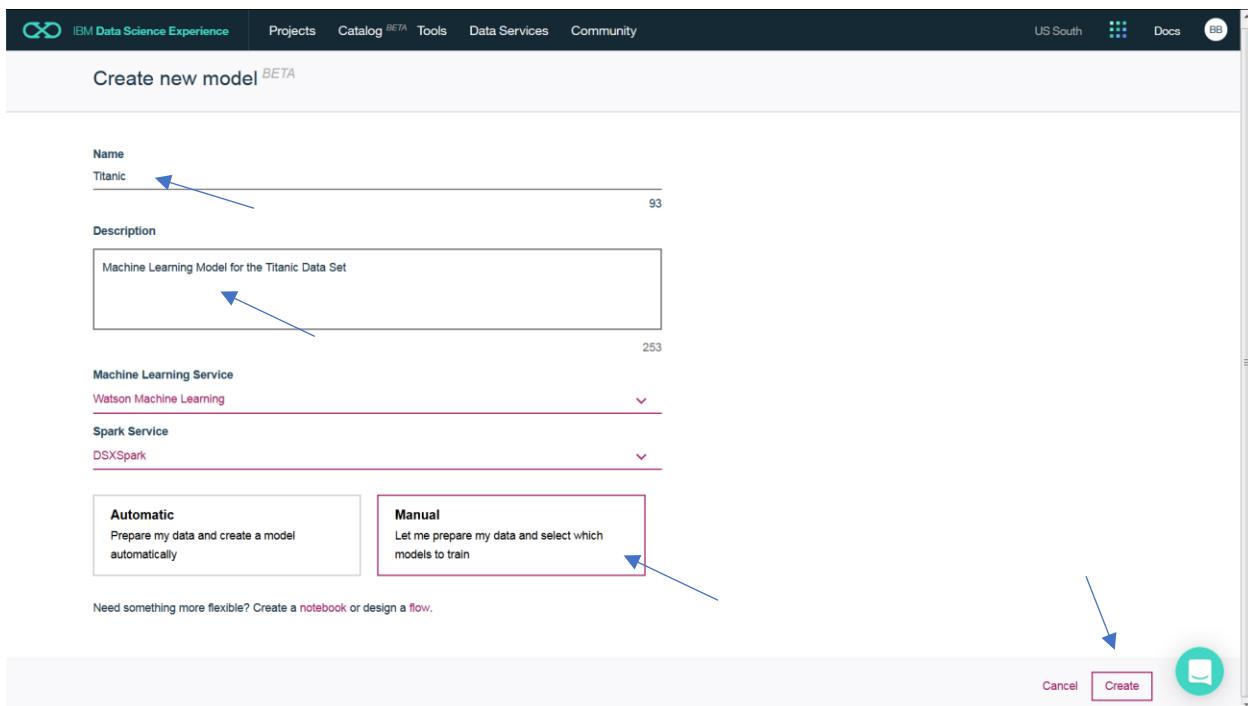
### 1. Click on the **Analytic Assets** Tab

The screenshot shows the IBM Data Science Experience interface. The top navigation bar includes File, Edit, View, History, Bookmarks, Tools, and Help. Below the bar, there are several tabs: Overview, Tutorial: Build a log... (active), Dashboard - IBM Bl... (active), IBM Data Science Exp... (active), Copy file from github..., Proof-of-Technology/..., and Proof-of-Technology/.... The main content area has a dark header with tabs: Overview, Analytics Assets (highlighted with a blue arrow), Data Assets, Bookmarks, Deployments, Collaborators, and Settings. Under the Analytics Assets tab, there are three sections: Notebooks, Data Assets, and Bookmarks. The Notebooks section shows two entries: 'Machine Learning with Spark - Joel' and 'Connect and Interact with DashDB'. The Data Assets section shows three entries: 'BikeShare.csv', 'Female-human-trafficking.csv', and 'myDashDB'. The Bookmarks section shows one entry. On the right side, there is a sidebar titled 'Files' with a 'Drop file here or browse your files to add a new file' area, and a 'Find in Storage' section listing 'Female-human-trafficking.csv' and 'BikeShare.csv'. The bottom of the screen shows a Windows taskbar with various icons and the time 11:34 PM.

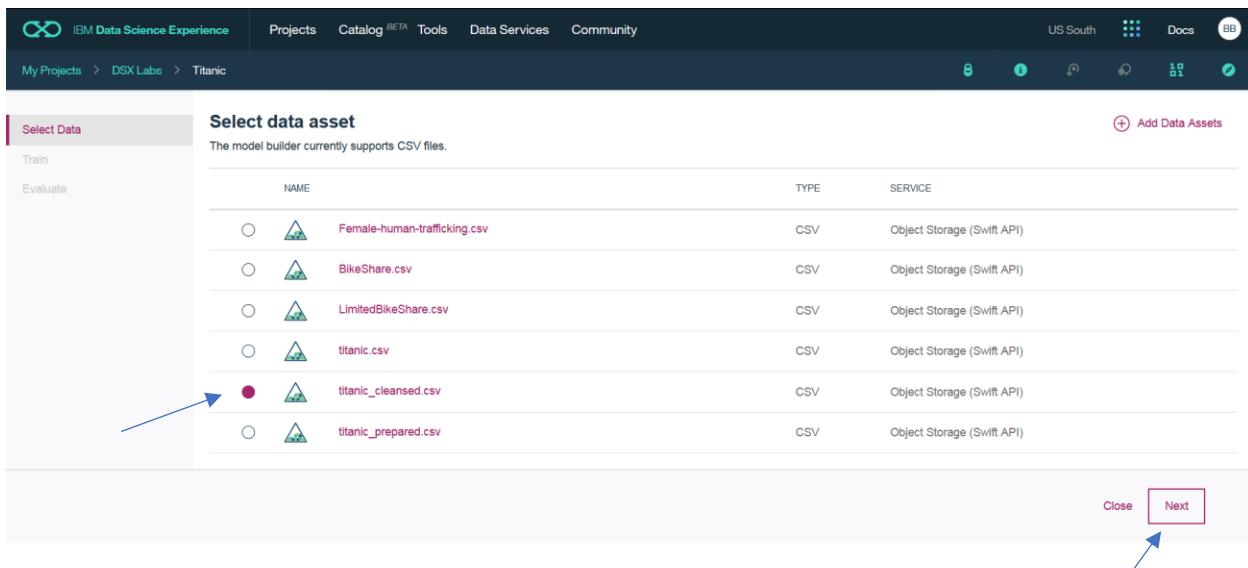
### 2. Click on **add models**.

The screenshot shows the same IBM Data Science Experience interface as the previous one, but with the Models section highlighted. The Models section is located under the Analytics Assets tab. It contains a table with columns: NAME, SHARED, STATUS, LANGUAGE, LAST EDITOR, LAST MODIFIED, and ACTIONS. There are two entries: 'Machine Learning with Spark - Joel' and 'Connect and Interact with DashDB'. Above the Models table, there is a search bar labeled 'Find in My Notebooks'. To the right of the Models table, there is a blue arrow pointing to a red '+' icon with the text '+ add models'. The bottom of the screen shows a Windows taskbar with various icons and the time 8:24 AM.

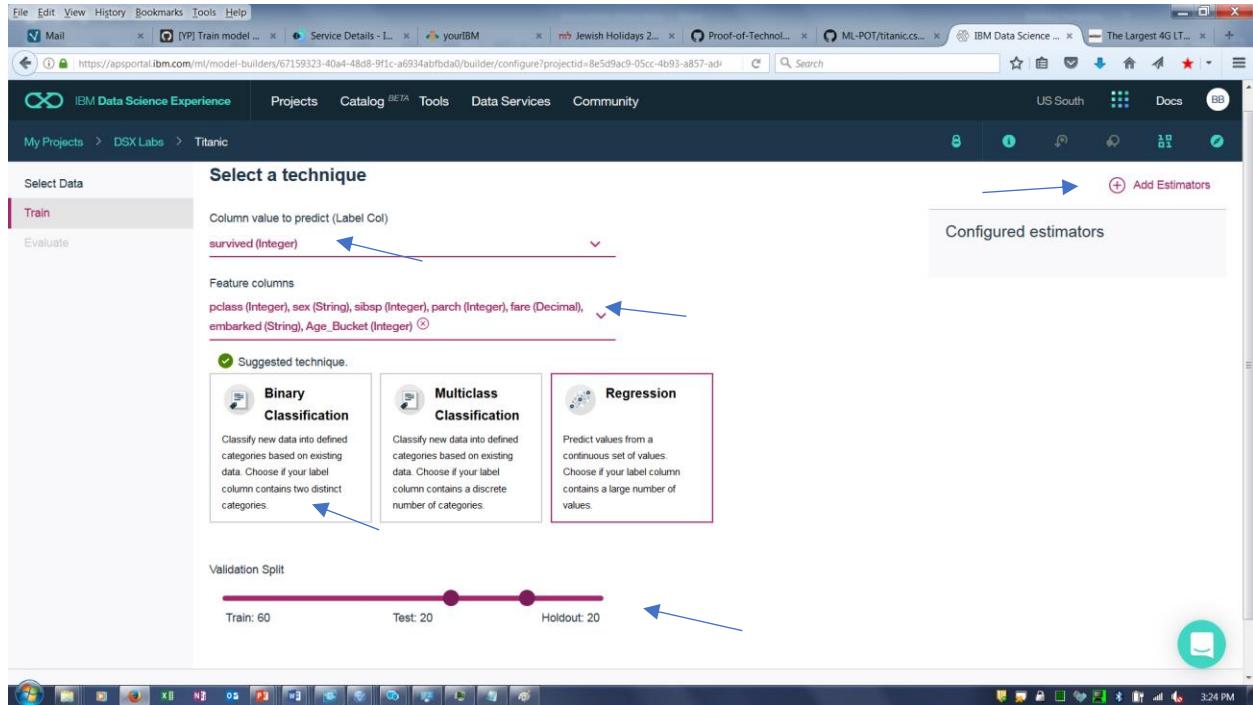
3. Enter the Model **Name**, **Description**, Select **Manual**, and click on **Create**.



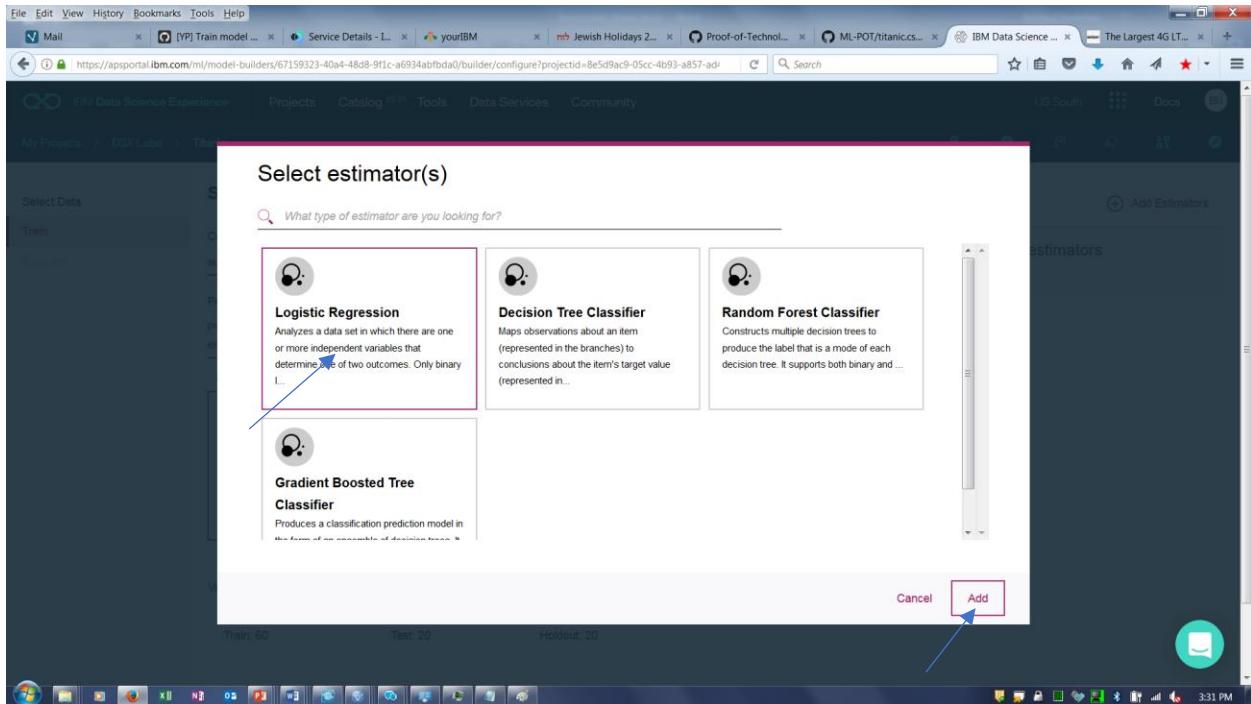
4. Click on the titanic\_cleansed.csv and click on **Next**



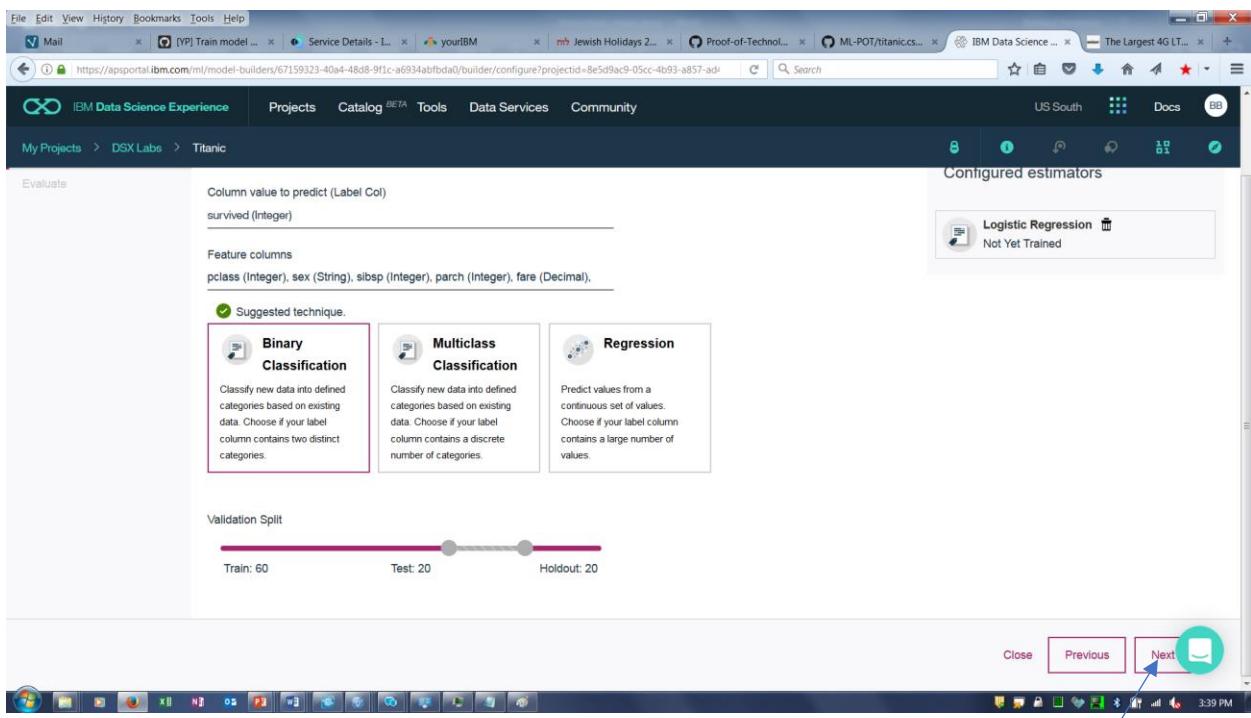
5. For **Column value to predict (Label Col)** select **survivor**. For **Feature columns** select the following features (**pclass, sex, sibsp, parch, fare, embarked, Age\_Bucket**) . Click on the **Binary Classification Box** (which is suggested by the service). Adjust the **Validation Split** as desired. Click on **Add Estimators** to add the specific models to use.



6. Select **Logistic Regression**. You can select more if you wish to see the results of multiple models. Select **Add**.



## 7. Select the Next button.



8. The system trains and evaluates each model. If more than one model was selected, the models would be listed in descending order of quality with the best result at the top.  
Note: if a model fails to run (rare, but happens), select Previous, delete that model and re-add it. Then run again. Click on **Logistic Regression** and then click **Save**.

Select model

	ESTIMATOR TYPE	STATUS	PERFORMANCE	AREA UNDER ROC CURVE	AREA UNDER PR CURVE	LAST EVALUATION	ACTIONS
<span style="color: red;">●</span>	LogisticRegression	Trained & Evaluated	Good	0.81287	0.81673	26 Aug 2017, 6:29 PM	...

Close Previous Save

9. The system displays the model training summary.

Titanic-WML

Overview Evaluation Deployments Test

**Summary**

Machine learning service	WML Instance
Runtime environment	spark-2.0
Training date	9 Oct 2017, 8:00 PM
Label column	survived
Latest version	6caeefdb-5f31-4c33-9908-3ecf10a29cf1
Model builder details	<a href="#">View</a>

**Input Schema**

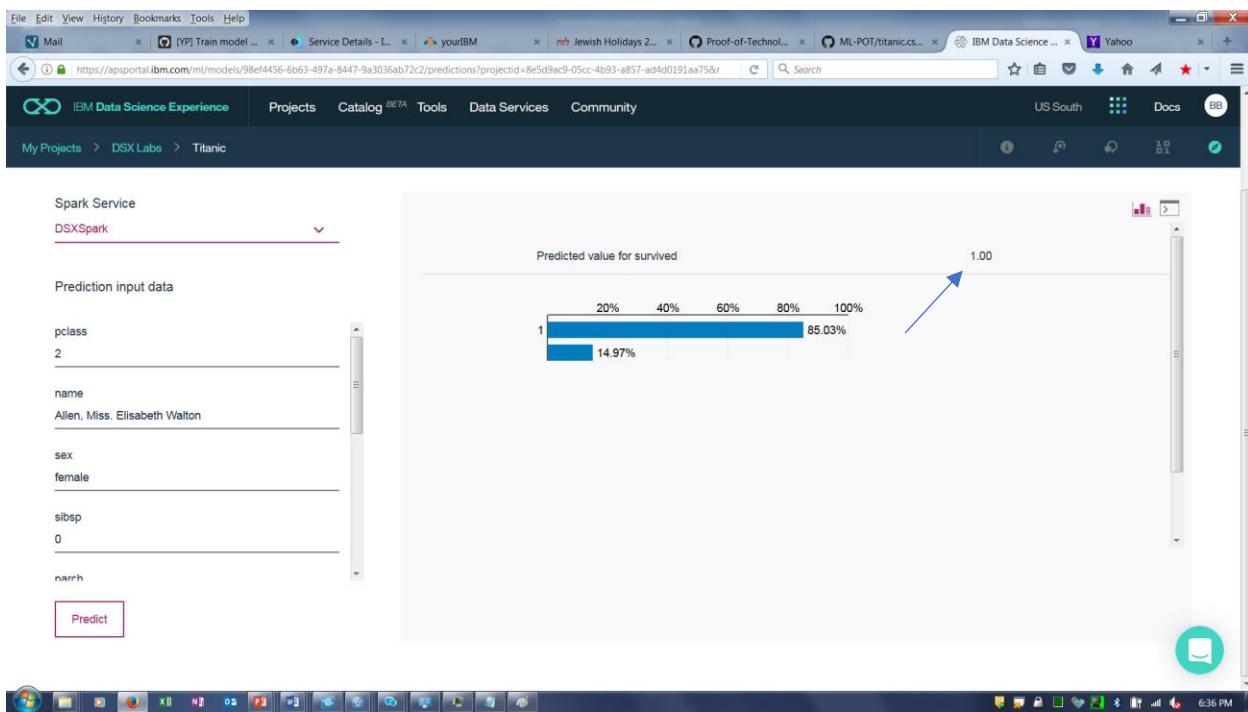
10. Select **Test** and then **Predict**

The screenshot shows the IBM Data Science Experience interface. At the top, the navigation bar includes 'My Projects > DSX Labs > Titanic'. Below the navigation is a 'Spark Service' dropdown set to 'DSXSpark'. The main area is titled 'Prediction input data' and contains the following fields:

- pclass: 2
- name: Allen, Miss. Elisabeth Walton
- sex: female
- sibsp: 0
- narrh

A blue arrow points from the 'Predict' button at the bottom left to the 'Predict' button on the right side of the interface.

11. The prediction for survivor is displayed along with the confidence in the prediction.



## Step 4: Deploying a Model

We can deploy the model to enable applications to invoke it via an API call.

1. Select the **Deployments** Tab
2. Scroll down to the **Add Deployments** option. Click on **Add Deployments**

The screenshot shows the IBM Data Science Experience web interface. At the top, there's a navigation bar with links for 'Projects', 'Tools', 'Data Services', and 'Community'. On the right side of the header, there are buttons for 'US South', 'Docs', and a user profile icon. Below the header, the URL 'My Projects > LMCO > Titanic-WML.' is visible. The main content area is titled 'Titanic-WML' with a subtitle '(WML)'. There are tabs for 'Overview', 'Evaluation', 'Deployments' (which is currently selected), and 'Test'. Under the 'Deployments' tab, there's a table with columns 'NAME' and 'DEPLOYMENT TYPE'. A message below the table says 'Your model is not deployed.' In the top right corner of the deployment table area, there's a button labeled '+ Add Deployment' with a blue arrow pointing to it.

3. Select Online for **Deployment Type**, enter Titanic\_Deployment for **Name**, and click on **Deploy**.

This screenshot shows a 'Deploy model' dialog box overlaid on the IBM Data Science Experience interface. The dialog has a title 'Deploy model'. Inside, there are fields for 'Deployment Type' (set to 'Online') and 'Name' (set to 'Titanic\_Deployment'). At the bottom right of the dialog, there are 'Close' and 'Deploy' buttons, with the 'Deploy' button highlighted by a red box and a blue arrow pointing to it. The background shows the main interface with various tabs like 'Mail', 'Service Details', and 'Jewish Holidays 2...', and a search bar at the top.

4. The system responds with an acknowledgement that the model was successfully deployed. Click on **Titanic\_Deployment** to test the deployed API.

The screenshot shows the IBM Data Science Experience web interface. At the top, there are multiple tabs open in a browser, including 'Mail', 'Service Details - L...', 'yourIBM', 'Jewish Holidays 2...', 'Proof-of-Tech...', 'ML-POT/titanic.cs...', 'IBM Data Science ...', and 'Yahoo'. Below the tabs, the main interface has a dark header with 'IBM Data Science Experience' and navigation links for 'Projects', 'Catalog BETA', 'Tools', 'Data Services', and 'Community'. The main content area shows a deployment summary for a project named 'Titanic'. A message at the top says 'Your deployment was successfully created.' A blue arrow points to this message. Below it is a table with deployment details:

Model builder details	<a href="#">View</a>
Training data schema	<a href="#">View</a>
Input data schema	<a href="#">View</a>
Runtime environment	spark-2.0
Training date	26 Aug 2017, 6:30 PM

Below this is a section titled 'Deployments' with a sub-section for 'Titanic\_Deployment'. A blue arrow points to the deployment name 'Titanic\_Deployment'. The table for deployments shows:

NAME	DEPLOYMENT TYPE	ACTIONS
Titanic_Deployment	Online	...

5. The system displays information about the deployed service including the endpoint to invoke by an application (e.g. web application predicting survival). Click on **Test API** to test out the API.

The screenshot shows the same IBM Data Science Experience interface as the previous one, but with a different focus. The 'Deployment Details' section is visible, and the 'Test API' tab is highlighted with a blue arrow pointing to it. The 'Details' tab is also visible. The deployment details table includes the following information:

Name	Titanic_Deployment
Type	online
Scoring End Point	<a href="https://ibm-watson-ml.mybluemix.net/v3/wml_instances/374817e5-8365-42da-a434-cb20e3d1fba4/published_models/98ef4456-6b63-497a-8447-9a3036ab72c2/deployments/00c07a3c-61ec-45ef-8cdde-213c18f1b448/online">https://ibm-watson-ml.mybluemix.net/v3/wml_instances/374817e5-8365-42da-a434-cb20e3d1fba4/published_models/98ef4456-6b63-497a-8447-9a3036ab72c2/deployments/00c07a3c-61ec-45ef-8cdde-213c18f1b448/online</a>
Status	ACTIVE
Associated Model	Titanic

6. Enter values for the input fields and then click on **Predict**. Note that the values inputted for any of the fields not included in the model parameters (e.g. name) will not affect the prediction.

My Projects > DSX Labs > Titanic\_Deployment

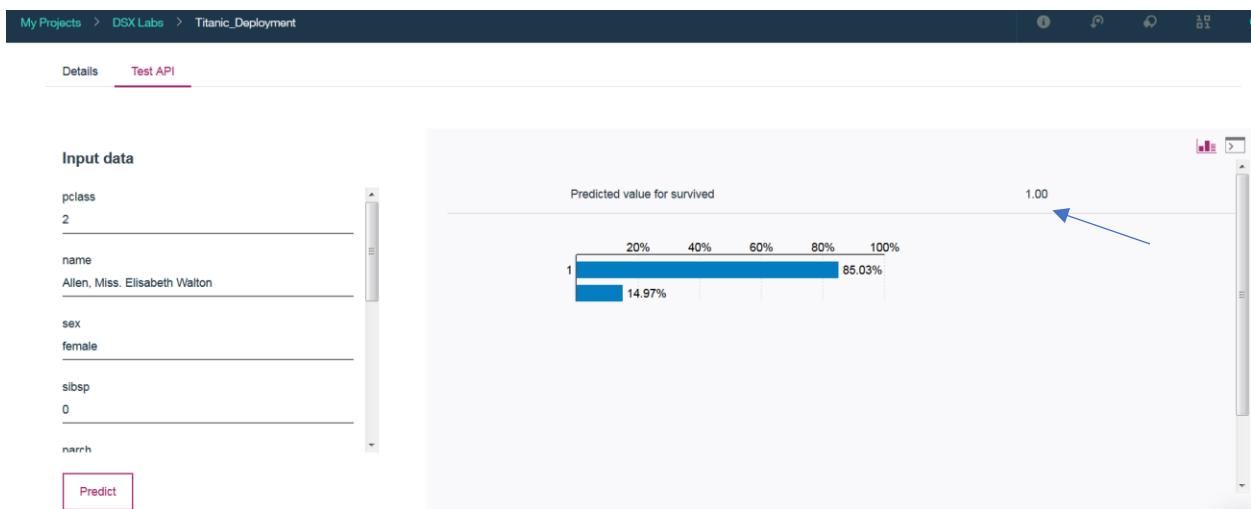
Details Test API

**Input data**

pclass	2
name	Allen, Miss. Elisabeth Walton
sex	female
sibsp	0
narch	

**Predict**

7. The predicted result is returned.



## Step 5: Deploy a simple web front-end to invoke the Watson Machine Learning service

This section will provide an example of a simple Python Flask front-end that invokes the Titanic scoring API demonstrating embedding machine learning in a web app. You will click on a link below that will deploy the sample Python web application into your Bluemix account. A toolchain will be set up for continuous delivery of the application. The application code will be cloned from a public Git repository into a private Git repo in your account that will be set up as part of the toolchain. Each time you commit changes to the repo, the app will be built and deployed.

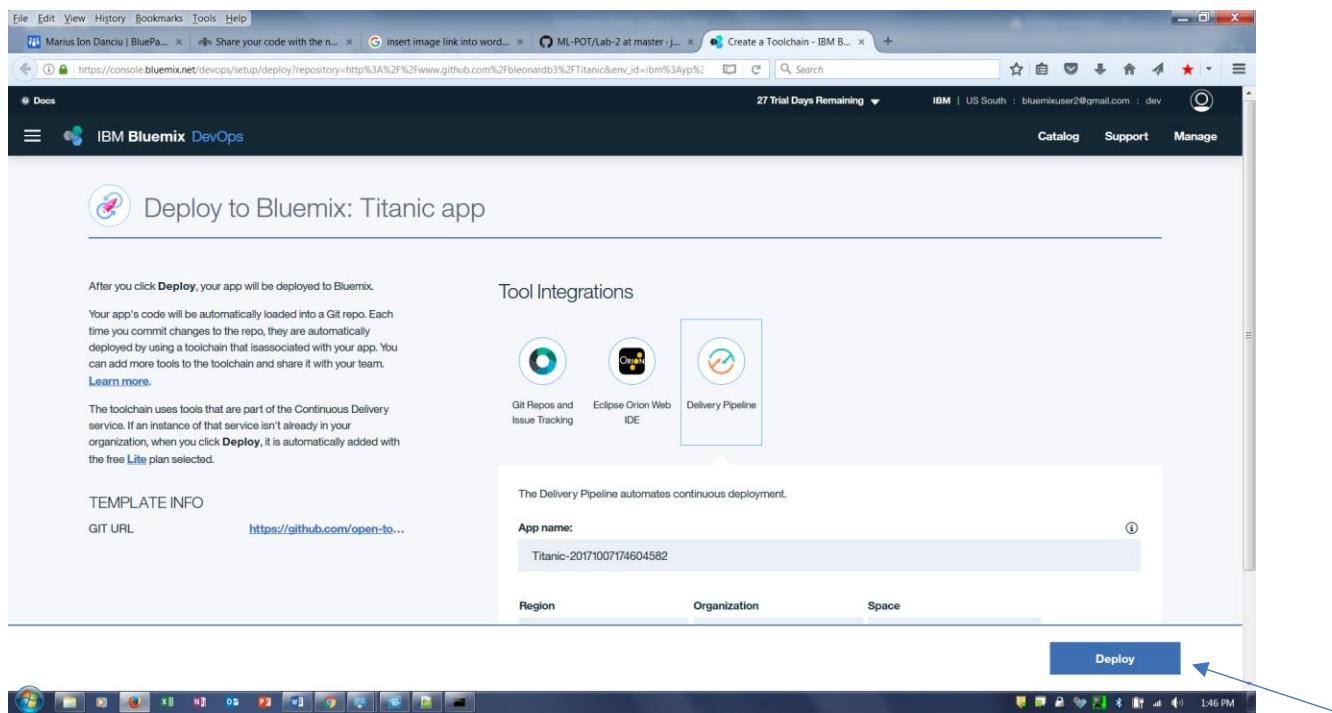
The toolchain uses tools that are part of the Continuous Delivery service. If an instance of that service isn't already in your organization, when you click **Deploy**, it is automatically added with the free [Lite](#) plan selected.

You will need to customize the application to provide the credentials for your Watson Machine Learning service, and to provide the scoring endpoint.

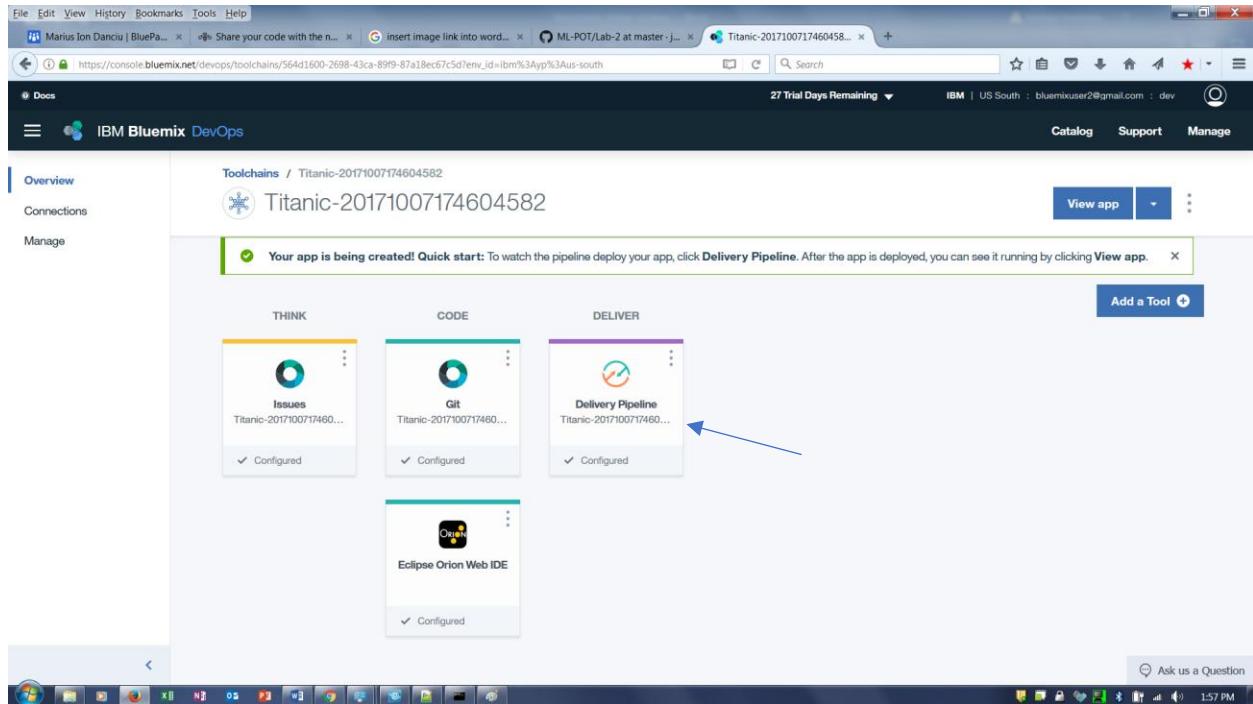
1. Click on the following to deploy a sample Python Flash web application into your Bluemix account. Note you may get a message to sign in to Bluemix.

[Deploy to Bluemix](#)

2. SCROLL DOWN TO WHERE THE “ORGANIZATION” and “SPACE” IS SHOWN. MAKE SURE THAT YOU UPDATE THE “SPACE” (CLICKING ON DOWN ARROW) TO BE THE SAME WHERE THE WATSON MACHINE LEARNING SERVICE IS DEPLOYED. CHANGE THE DEFAULT (“DEV”) TO THE APPROPRIATE SPACE. Click on the Deploy button.

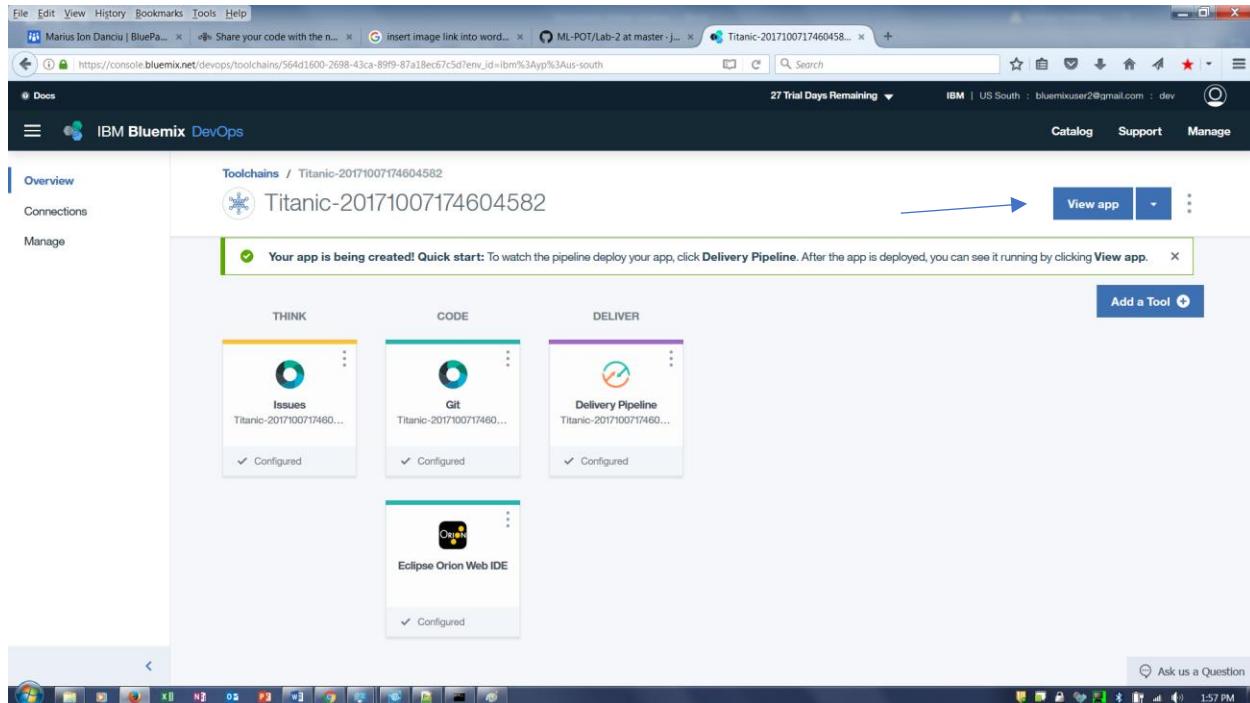


3. Your app is being created! To watch the pipeline deploy your app, click **Delivery Pipeline**.

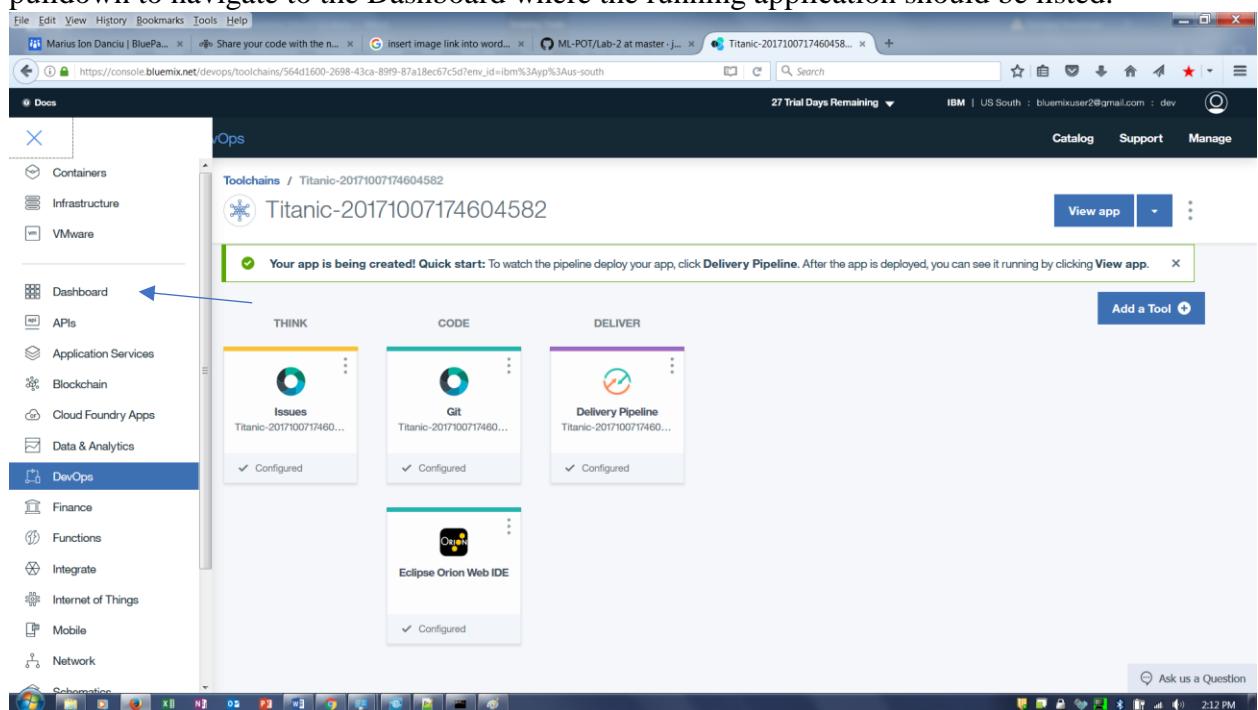


4. After the app is deployed successfully (should say Deployed passed now in the Deploy stage), return to the Delivery Pipeline by clicking on the vertical ellipse and click on View Toolchain.

5. You can see the running app by clicking **View app**. The web form collecting the Titanic passenger data should appear. Note that the application is not functional until we connect it to the Watson Machine Learning service so if you Submit you will get an error!



6. Close the Titanic prediction app tab, and click on the  icon and Dashboard in the pulldown to navigate to the Dashboard where the running application should be listed.



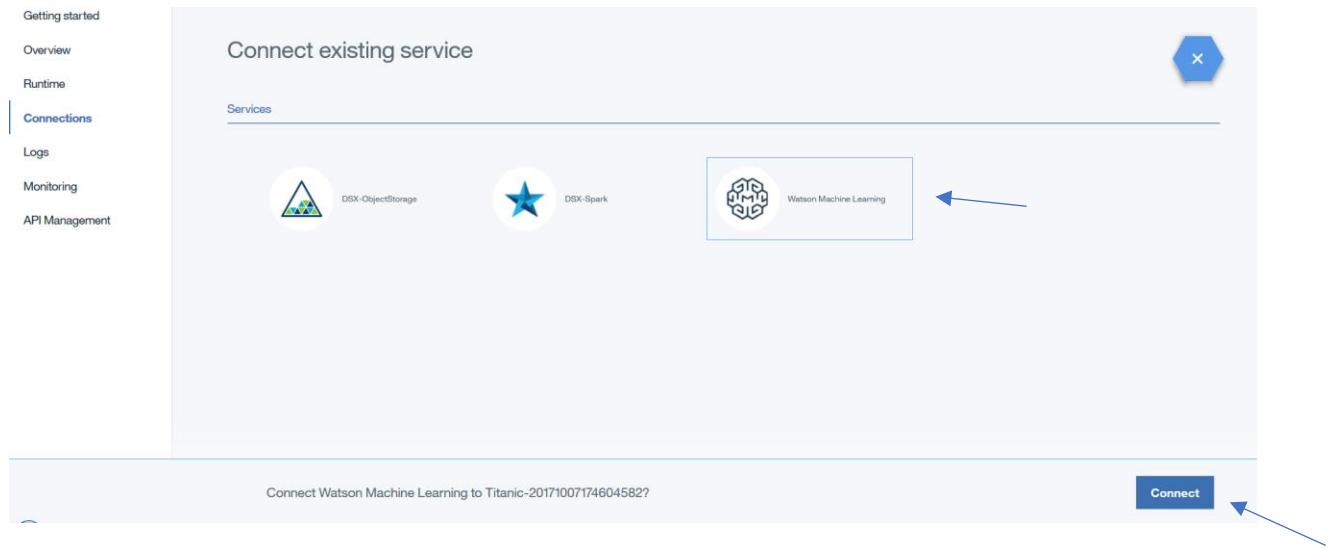
7. We are now going to connect the application to the Watson Machine Learning service that was created earlier. Click on the application name.

The screenshot shows the IBM Bluemix Dashboard. In the top navigation bar, there are tabs for 'Catalog', 'Support', and 'Manage'. Below the navigation, there's a search bar labeled 'Search Items' and a 'Create' button. The main area is divided into two sections: 'Cloud Foundry Apps (1)' and 'Services (5)'. The 'Cloud Foundry Apps' section shows one app named 'Titanic-20171007174604582' with a route 'titanic-20171007174604582-jackleg-pirogue.mybluemix.net', 1 instance, 128 MB memory, and it is running. The 'Services' section lists five services: Continuous Delivery, Data Science Experience-gz, DSX-ObjectStorage, DSX-Spark, and Watson Machine Learning. A blue arrow points from the text 'Click on the application name.' to the application name 'Titanic-20171007174604582' in the 'Cloud Foundry Apps' table.

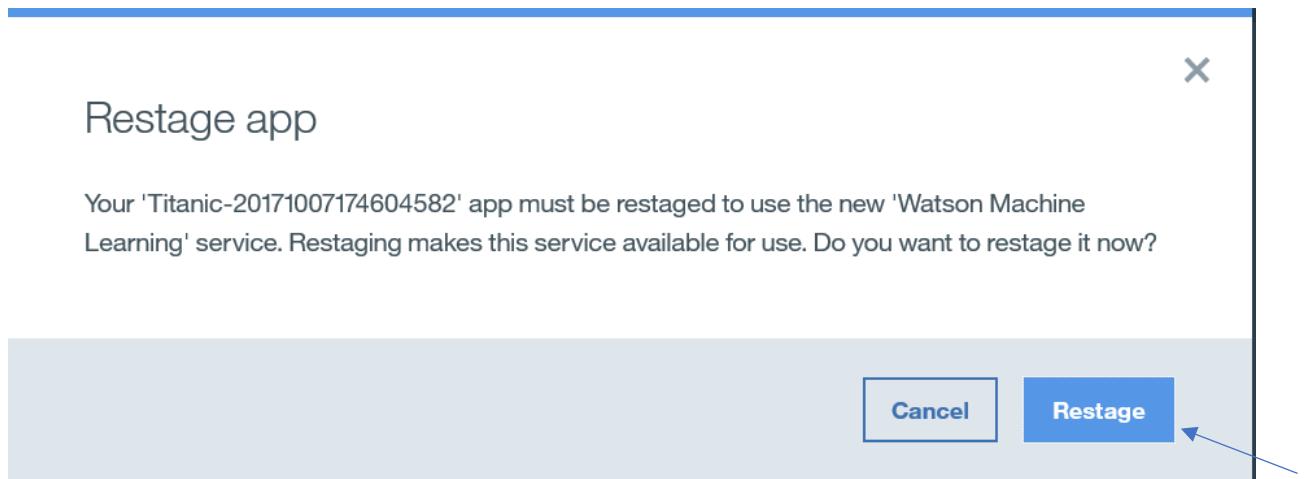
8. Scroll down until you see the Connections panel. Click on Connect Existing.

The screenshot shows the 'Application Details - IBM ...' page for the 'Titanic-20171007174604582' app. On the left, there's a sidebar with options like 'Getting started', 'Overview' (which is selected), 'Runtime', 'Connections', 'Logs', 'Monitoring', and 'API Management'. The main content area displays the app's configuration: buildpack (.py), 1 instance, 128 MB memory per instance, and a total of 128 MB allocation. Below this, the 'Connections' panel is visible, showing a message 'No services are connected to this app. You can create or bind a service:' with two buttons: 'Connect new' and 'Connect existing'. A blue arrow points from the text 'Click on Connect Existing.' to the 'Connect existing' button. To the right of the connections panel, there's a 'Runtime cost' section showing '\$0.00' for both current and estimated charges, with a note that costs exclude connected services and a 'View full usage details' button.

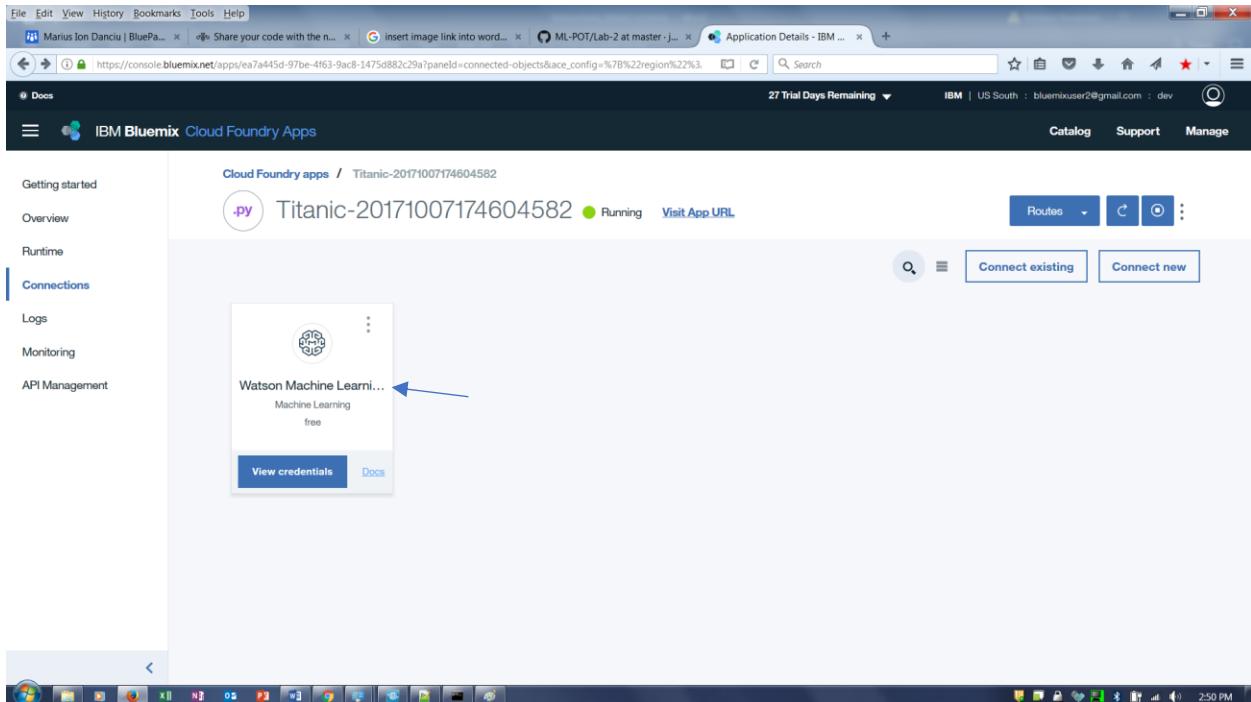
9. You should see at least 3 services listed. Two services are automatically provisioned for the Data Science Experience (Spark service, and Object Storage service). Click on the **Watson Machine Learning** service, and then click on **Connect**.



10. You will get a pop up that asks to Restage the application. Click on **Restage**.



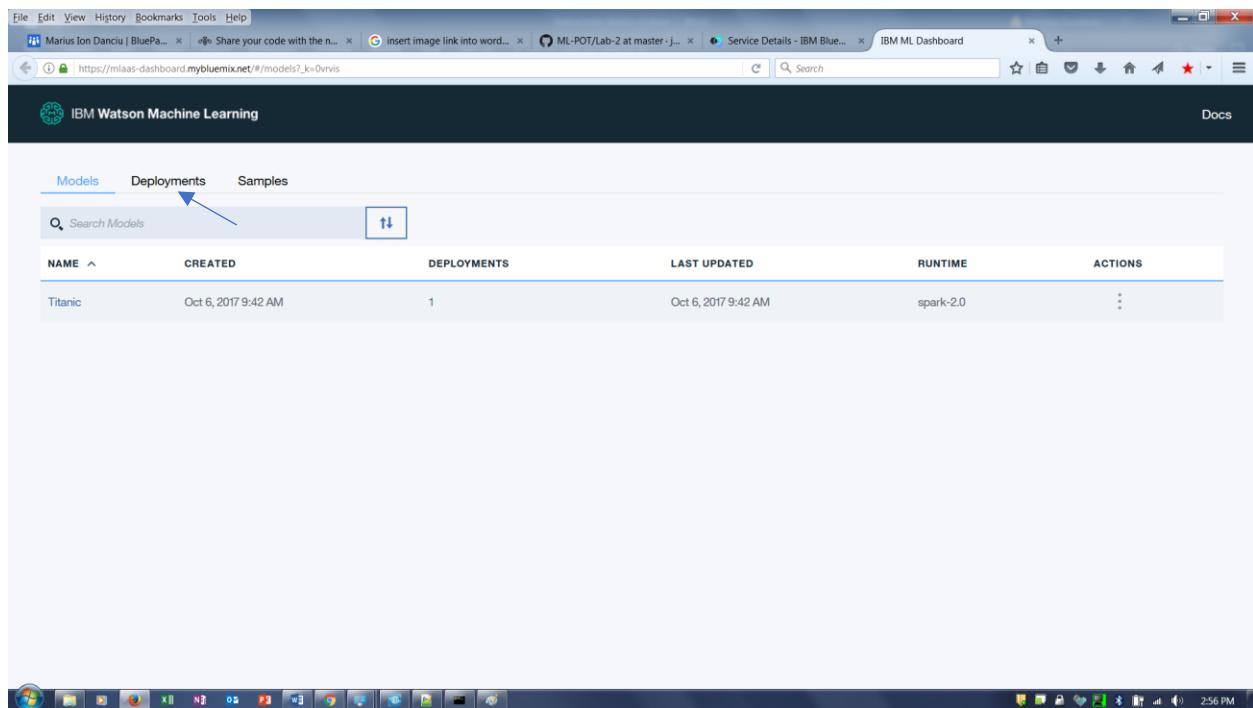
11. We now need to copy the scoring endpoint to the clipboard. Click on the **Watson Machine Learning** service.



## 12. Click on the Watson Machine Learning service **Launch Dashboard**.

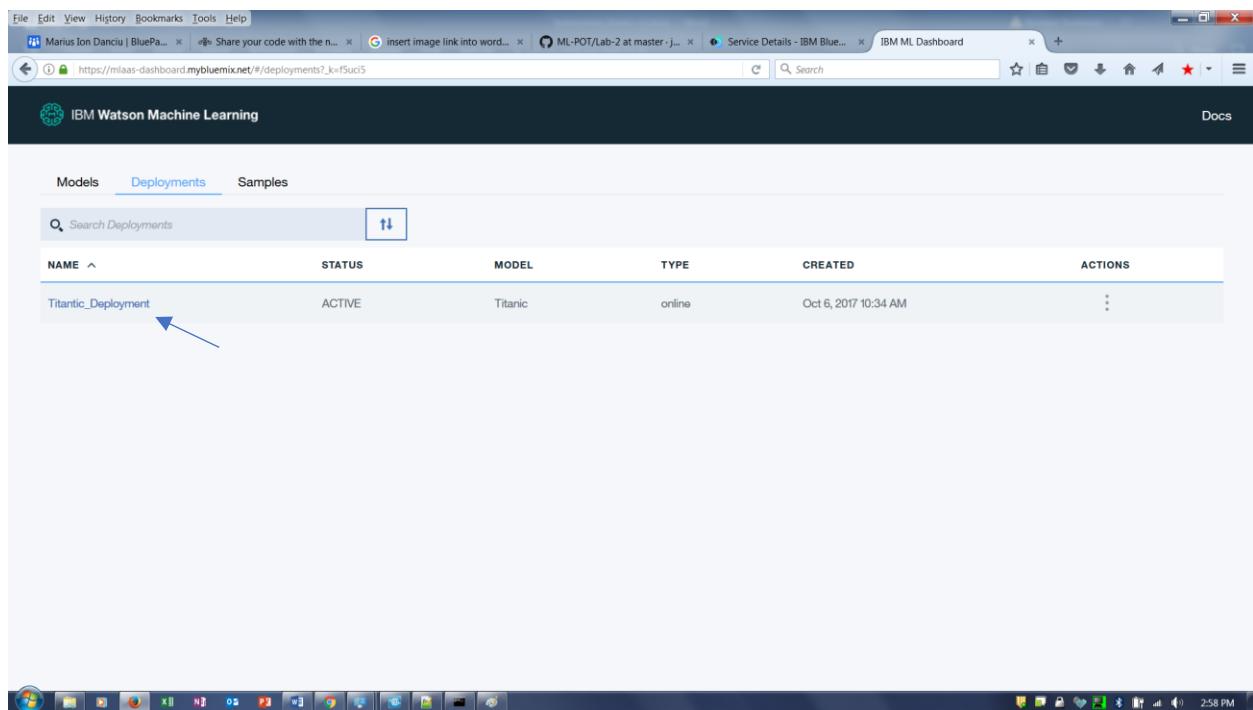
The screenshot shows the IBM Bluemix Data & Analytics dashboard. On the left, a sidebar has 'Manage' selected, with sub-options like 'Service credentials', 'Plan', and 'Connections'. The main content area is titled 'Watson Machine Learning'. It features a 'Dashboards' section with a note about providing REST APIs for integration. Below this are two cards: 'SPSS Streams Service' and 'Watson Machine Learning'. Each card has a 'Launch Dashboard' button at the bottom. A blue arrow points to the 'Launch Dashboard' button under the 'Watson Machine Learning' card. At the bottom, there's a 'Sample Applications' section with a note about building own apps.

## 13. Click on the **Deployment** tab.



The screenshot shows the IBM Watson Machine Learning interface. At the top, there are several tabs: 'File', 'Edit', 'View', 'History', 'Bookmarks', 'Tools', and 'Help'. Below the tabs, the URL is https://mlaas-dashboard.mybluemix.net/#/models?\_k=0vrvis. The main header says 'IBM Watson Machine Learning' and 'Docs'. There are three tabs at the top: 'Models' (selected), 'Deployments' (with a blue arrow pointing to it), and 'Samples'. Below the tabs is a search bar with placeholder text 'Search Models' and a refresh icon. A table follows, with columns: NAME, CREATED, DEPLOYMENTS, LAST UPDATED, RUNTIME, and ACTIONS. One row is visible for 'Titanic', created on Oct 6, 2017, with 1 deployment, last updated on Oct 6, 2017, at 9:42 AM, using spark-2.0 runtime. The bottom of the screen shows a Windows taskbar with various icons.

14. Click on the Titanic Deployment.



The screenshot shows the same IBM Watson Machine Learning interface, but now the 'Deployments' tab is selected. The URL has changed to https://mlaas-dashboard.mybluemix.net/#/deployments?\_k=f5uc5. The main header and tabs remain the same. The table below has columns: NAME, STATUS, MODEL, TYPE, CREATED, and ACTIONS. One row is visible for 'Titanic\_Deployment', which is active, uses the 'Titanic' model, is an 'online' type, was created on Oct 6, 2017, at 10:34 AM, and has an ellipsis icon in the ACTIONS column. A blue arrow points to the 'Titanic\_Deployment' row. The bottom of the screen shows a Windows taskbar.

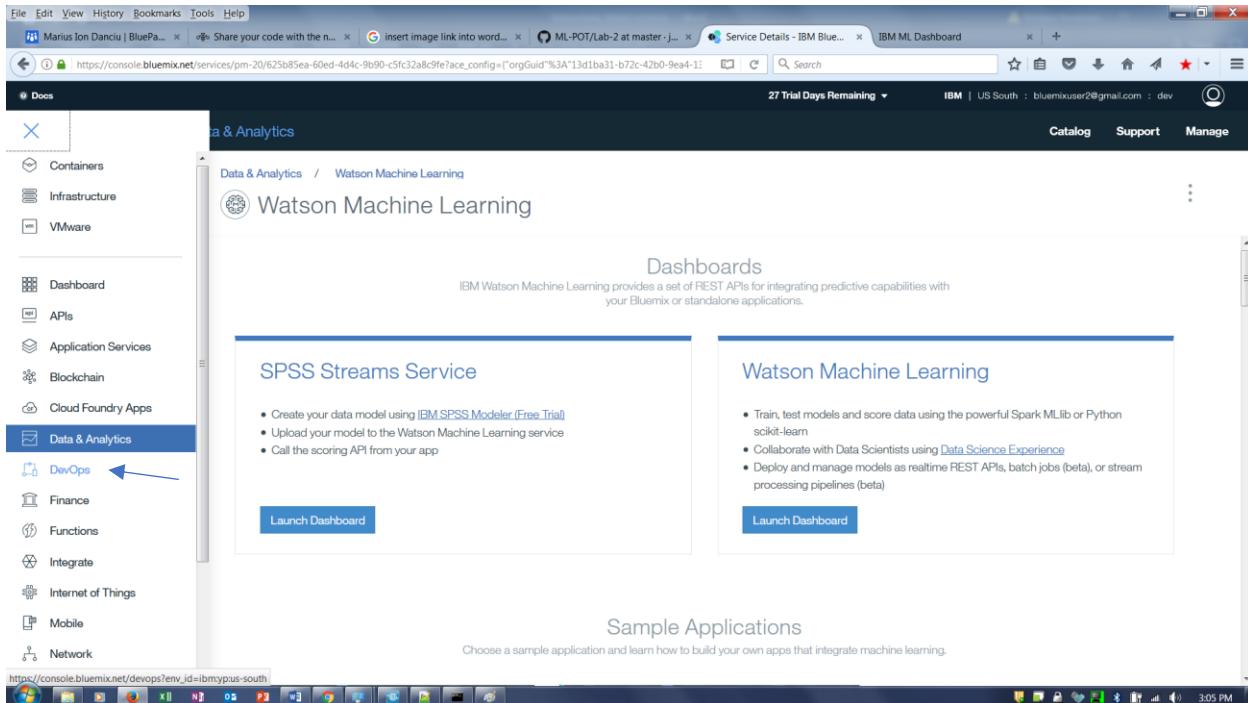
15. Scroll down to API details and copy the Scoring Endpoint value “http.....etc”

Deployment Name	Titanic_Deployment
Deployment Type	online
Deployment Id	7e591f82-e004-46e7-b35d-c9fb062ac8a
Deployment Status	ACTIVE
URL	<a href="https://ibm-watson-ml.mybluemix.net/v3/wml_instances/625b85ea-60ed-4d4c-9b90-c5fc32a8c9fe/published_models/0788b520-0f19-4f7c-9ee5-03a9f7ed899d">https://ibm-watson-ml.mybluemix.net/v3/wml_instances/625b85ea-60ed-4d4c-9b90-c5fc32a8c9fe/published_models/0788b520-0f19-4f7c-9ee5-03a9f7ed899d</a>
Model Name	Titanic
Model URL	<a href="https://ibm-watson-ml.mybluemix.net/v3/wml_instances/625b85ea-60ed-4d4c-9b90-c5fc32a8c9fe/published_models/0788b520-0f19-4f7c-9ee5-03a9f7ed899d">https://ibm-watson-ml.mybluemix.net/v3/wml_instances/625b85ea-60ed-4d4c-9b90-c5fc32a8c9fe/published_models/0788b520-0f19-4f7c-9ee5-03a9f7ed899d</a>
Created	Oct 6, 2017 10:34 AM
Last Updated	Oct 6, 2017 10:34 AM

API Details

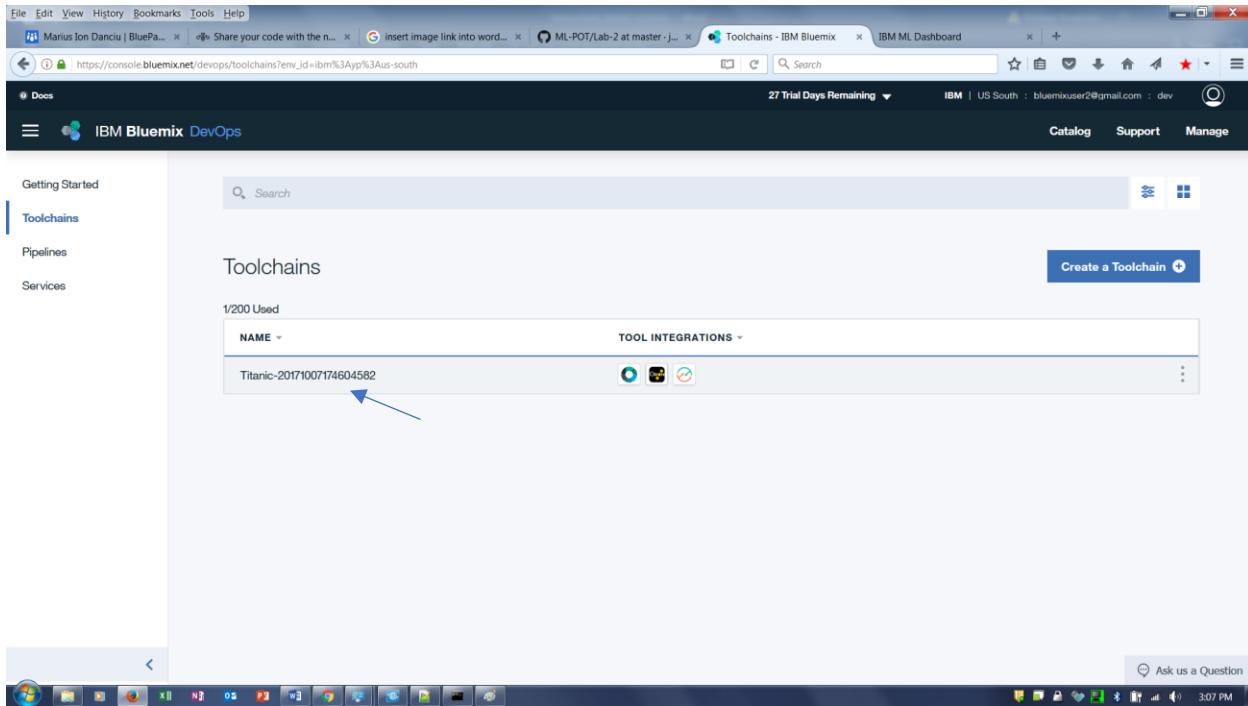
Scoring Endpoint: [https://ibm-watson-ml.mybluemix.net/v3/wml\\_instances/625b85ea-60ed-4d4c-9b90-c5fc32a8c9fe/published\\_models/0788b520-0f19-4f7c-9ee5-03a9f7ed899d](https://ibm-watson-ml.mybluemix.net/v3/wml_instances/625b85ea-60ed-4d4c-9b90-c5fc32a8c9fe/published_models/0788b520-0f19-4f7c-9ee5-03a9f7ed899d)

16. Close the Deployment screen, and Select the Service Details tab. Click on the  icon, and click on DevOps in the pulldown to navigate to the Toolchain.

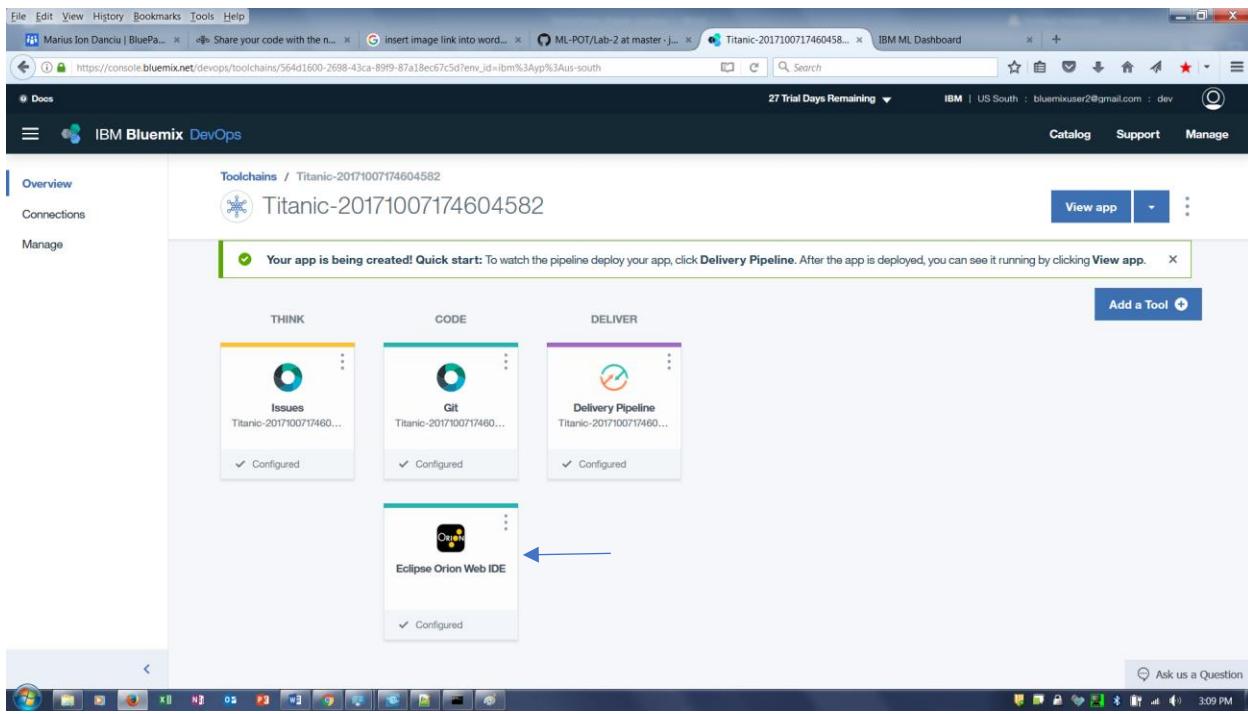


The screenshot shows the IBM ML Dashboard interface. At the top, there are several tabs: 'Service Details - IBM Blue...', 'ML-POT/Lab-2 at master', 'IBM ML Dashboard', and others. Below the tabs, there's a search bar and navigation icons. The main content area is titled 'Watson Machine Learning' and contains sections for 'Dashboards', 'SPSS Streams Service', and 'Watson Machine Learning'. The left sidebar has a 'Data & Analytics' section with a 'DevOps' item highlighted, indicated by a blue arrow. The status bar at the bottom shows the URL 'https://console.bluemix.net/devops?env\_id=ibmypus-south' and the time '3:05 PM'.

17. We are now going to paste the scoring endpoint into the application code. Click on the Toolchain.



### 18. Click on the Eclipse Orion Web IDE.



### 19. Click on the Titanic\_v1 file.



This repository contains a Python Flask program that invokes a Watson Machine Learning service to predict survival based on passenger information from the Titanic.

Titanic-20171007174604582

README.md

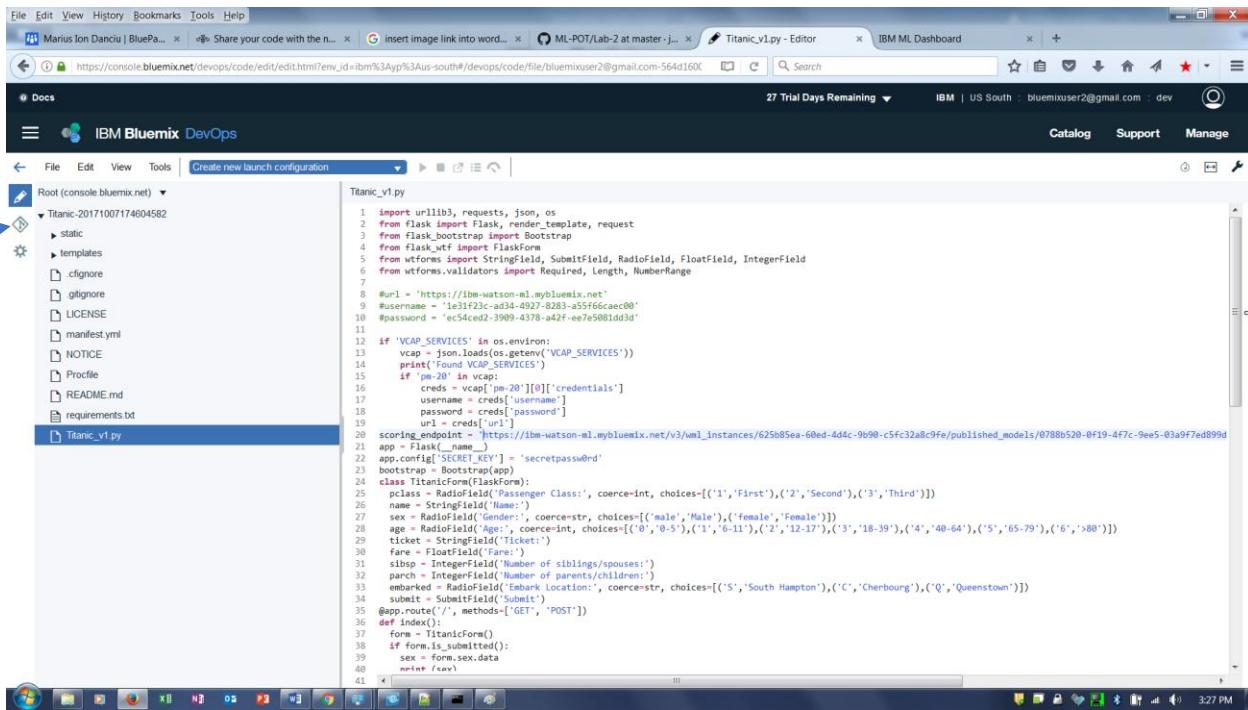
Titanic-20171007174604582

static	10/7/2017, 3:11:16 PM
templates	10/7/2017, 3:11:16 PM
.cignore	10/7/2017, 3:11:18 PM
.gitignore	10/7/2017, 3:11:18 PM
LICENSE	10/7/2017, 3:11:16 PM
manifest.yml	10/7/2017, 3:11:16 PM
NOTICE	10/7/2017, 3:11:16 PM
Profile	10/7/2017, 3:11:16 PM
README.md	10/7/2017, 3:11:16 PM
requirements.txt	10/7/2017, 3:11:16 PM
Titanic_v1.py	10/7/2017, 3:11:16 PM

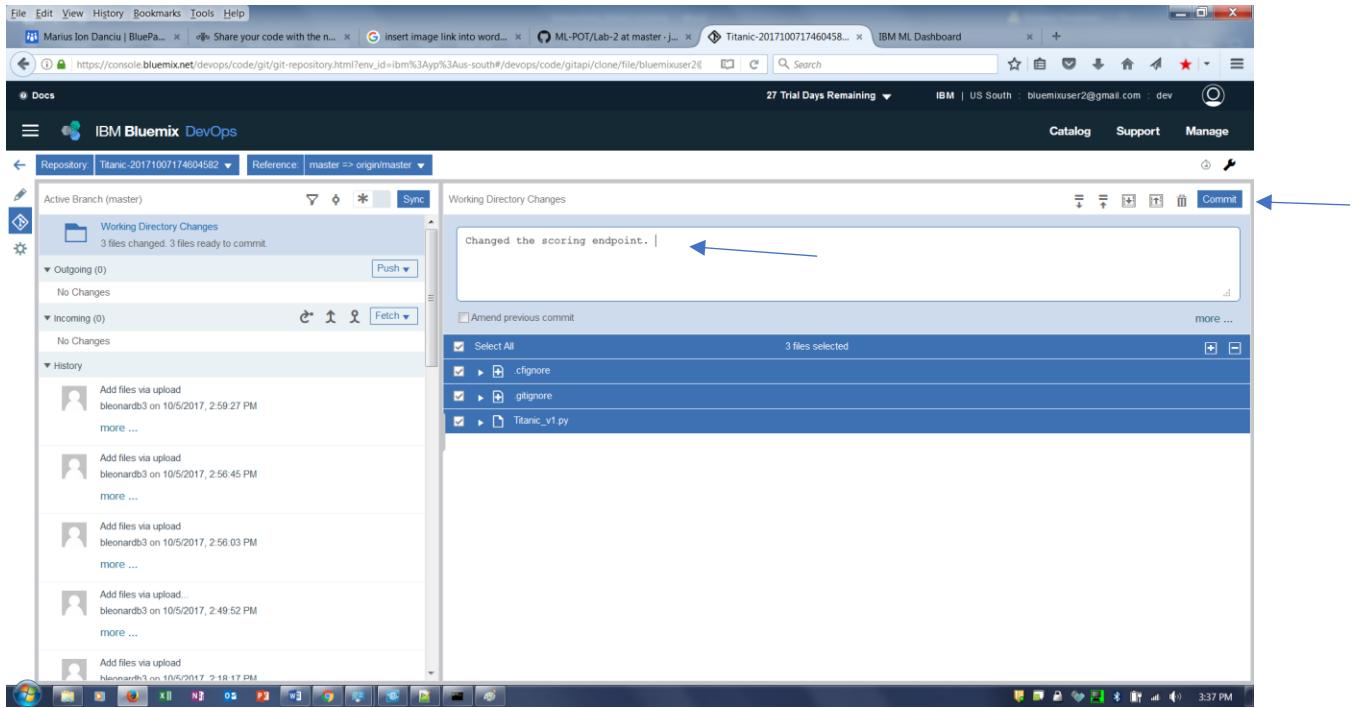
► Git

► JavaScript

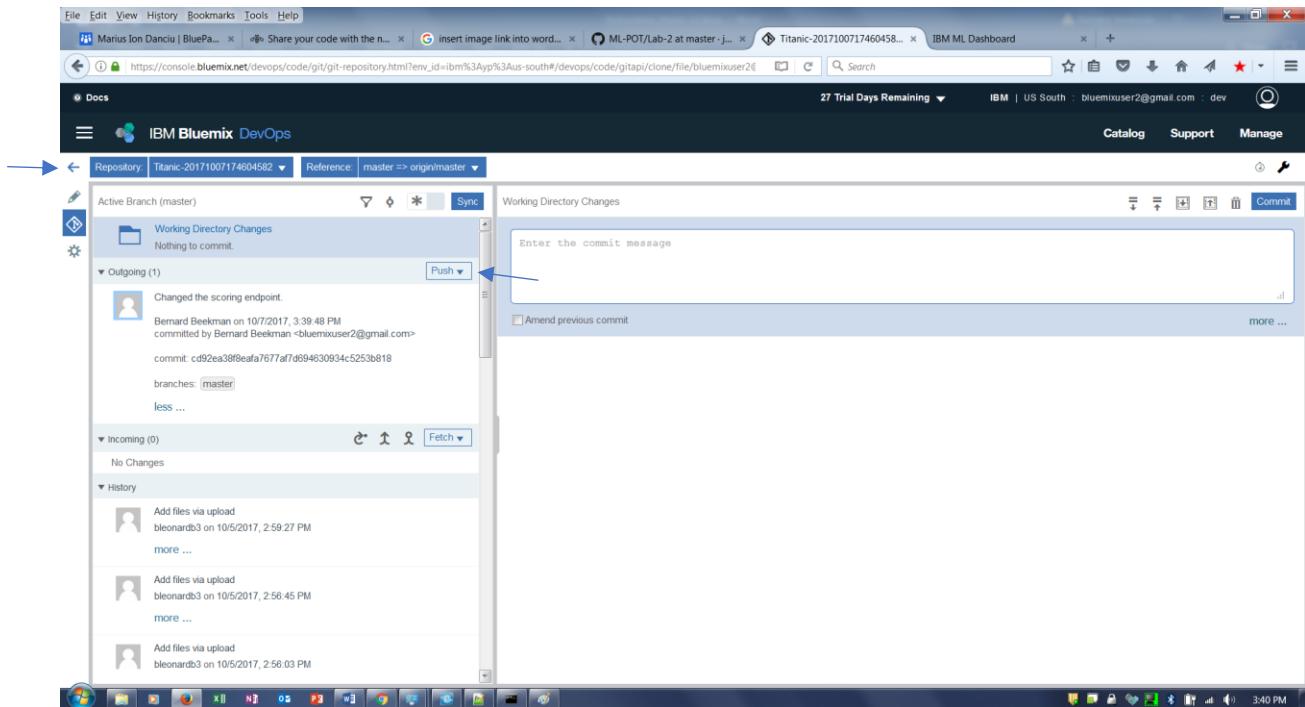
20. Look around line 20 for the “scoring endpoint =”. Select the current scoring endpoint (may want to use Shift-End to get to the end of the line. Make sure not to select the endpoint quote – if you do just make sure to put it back in). Enter Ctrl-V to paste the new scoring endpoint from your Watson Machine Learning service. Enter Ctrl-S to save the file. Then click on the  icon on the top left.



21. The next step is to commit the change to the repository. Enter “Changed the Scoring Endpoint” in the Enter Commit Message field, and then click on **Commit**.



22. Then click on **Push** to push the changes to the Git repo which will start the build and deploy of the application. Click on the left arrow to return to the Toolchain.



23. Click on the **Delivery Pipeline** to view status of the deployment as before. Once the Deployment status shows **Deploy passed now**, click on the vertical ellipsis and then click

on the **View Toolchain** option to return to the Toolchain screen. Click on the **View Apps** button. (see Steps 3,4,5 above as a reminder if necessary). The web form should appear. Enter data in all the fields and click on the **Submit** button. (the submit button is located at the bottom of the web form – you may need to scroll).

To determine the survival prediction,please enter the following:

Passenger Class:

First  
 Second  
 Third

Name: Bernie Beekman

Gender:

Male  
 Female

Number of siblings/spouses: 1

Number of parents/children: 1

Ticket: 1234

Fare: 23

Embarck Location:

South Hampton  
 Cherbourg  
 Queenstown

Age:

0-5  
 6-11  
 12-17

24. You should see something similar to the following depending on the values of the input fields that you entered. Click on the **Try Again!**, if you want to experiment with different inputs.

Titanic Prediction

```
prediction:survived
probability: 0.827966430684
```

[Try Again!](#)