

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

IBM Data Science Experience | Projects Tools Data Services Community

My Projects > LMCO

Add to project •

US South Docs JP

Overview Analytics Assets Data Assets Bookmarks Deployments Collaborators Settings

Project Information

Project Name
LMCO

Description
Project Description

96 3000

Cancel Save

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

NAME	SERVICE TYPE	PLAN
DSX-Spark	Spark	Personal

+ add associated service ^

- Amazon EMR Spark
- IBM Analytics Engine
- Machine Learning
- Spark

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

IBM Data Science Experience | Projects Tools Data Services Community

US South Docs JP

Add Machine Learning Service

Existing New

Machine Learning

IBM Watson Machine Learning is a full-service Bluemix offering that makes it easy for developers and data scientists to work together to integrate predictive capabilities with their applications. Built on IBM's proven SPSS analytics platform, Machine Learning allows you to develop applications that make smarter decisions, solve tough problems, and improve user outcomes.

Features

SPSS analytics platform features
SPSS streams management and deployment with realtime scoring and batch processing options.

Coming Soon! Integration with Data Science Experience
Visit <http://datascience.ibm.com>. Take advantage of Spark ML pipelines deployment - realtime, batch and streaming. Monitor API usage and performance of deployed models.

Pricing Plan: Monthly Process shown above reflect the: **United States**

Plan	Features	Pricing
Free	Service instance (5 models per instance) 5,000 predictions 5 compute hours	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

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 and 0 forks. The 'titanic_cleaned.csv' file is shown, which has 1045 lines, 1044 sloc, and 75.6 KB. The file was added by 'jpatter' 6 hours ago. Below the file information, there is a table of data. An arrow points to the 'Raw' button in the top right corner of the table's metadata section.

1	pclass	survived	name	sex	sibsp	parch	ticket	fare	embarked
2	1	1	Allen, 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 links for Overview, Tutorial, Dashboard, IBM Data Science Experience, Copy file from github..., Proof-of-Technology/, and Proof-of-Technology/. The main menu bar has options like File, Edit, View, History, Bookmarks, Tools, and Help.

The current view is 'My Projects > DSX Labs'. The interface is divided into several sections:

- Notebooks**: Shows two entries: 'Machine Learning with Spark - Joel' and 'Connect and Interact with DashDB'.
- Data Assets**: Shows two entries: 'Female-human-trafficking.csv' (CSV, Object Storage) and 'myDashDB' (Connection).
- Bookmarks**: Shows one entry: 'A comparison of logistic regression a...' (ARTICLE).

A blue arrow points to the '+ add data assets' button in the Data Assets section.

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

This screenshot shows the same IBM Data Science Experience interface as the previous one, but with a different set of data assets listed under the 'Data Assets' section:

- 'BikeShare.csv' (CSV, Object Storage)
- 'Female-human-trafficking.csv' (CSV, Object Storage)
- 'myDashDB' (Connection)

A blue arrow points to the 'Drop file here or browse your files to add a new file' input field, which is part of a sidebar panel labeled 'Files'.

Step 3: Create a Model to predict survival

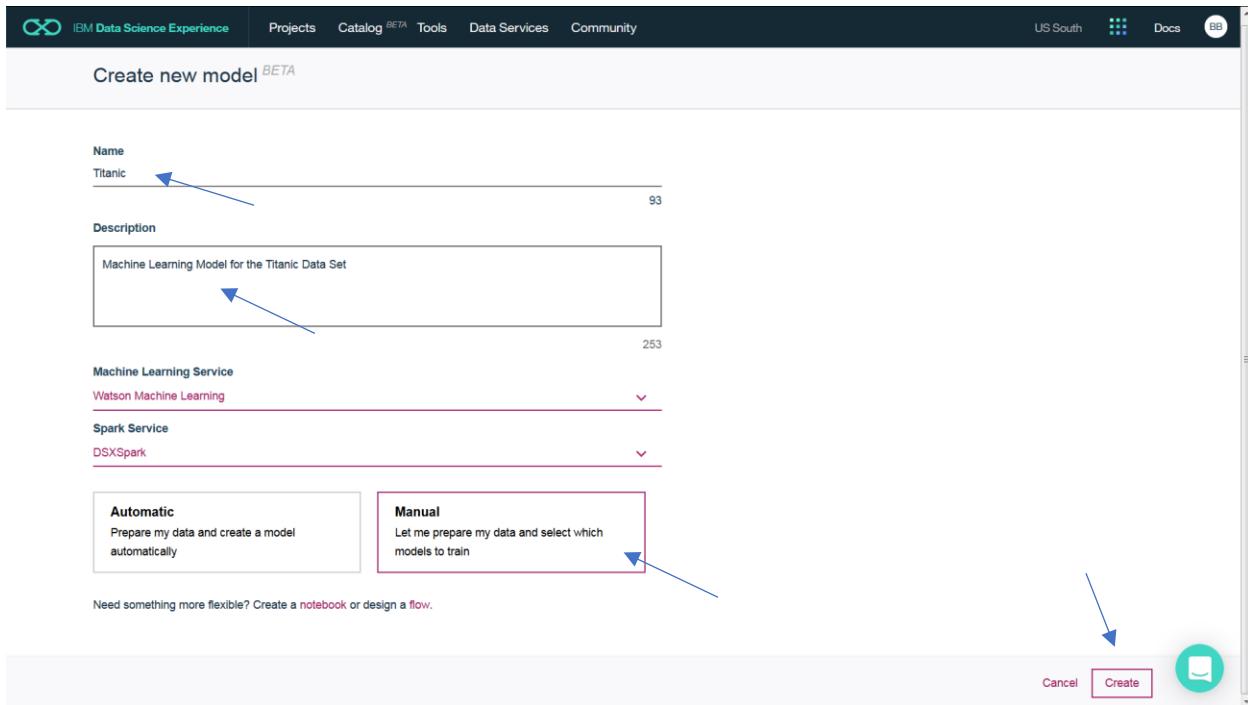
1. Click on the **Analytic Assets** Tab

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 (selected), Data Assets, Bookmarks, Deployments, Collaborators, and Settings. A red arrow points to the 'Analytics Assets' tab. On the left, there's a sidebar with sections: Notebooks (view all (2)), Data Assets (view all (3)), and Bookmarks (view all (1)). On the right, there's a 'Files' section with a 'Drop file here or browse' area, a 'Find in Storage' section listing 'Female-human-trafficking.csv' and 'BikeShare.csv', and a 'Connections' section. The bottom of the screen shows a Windows taskbar with various icons.

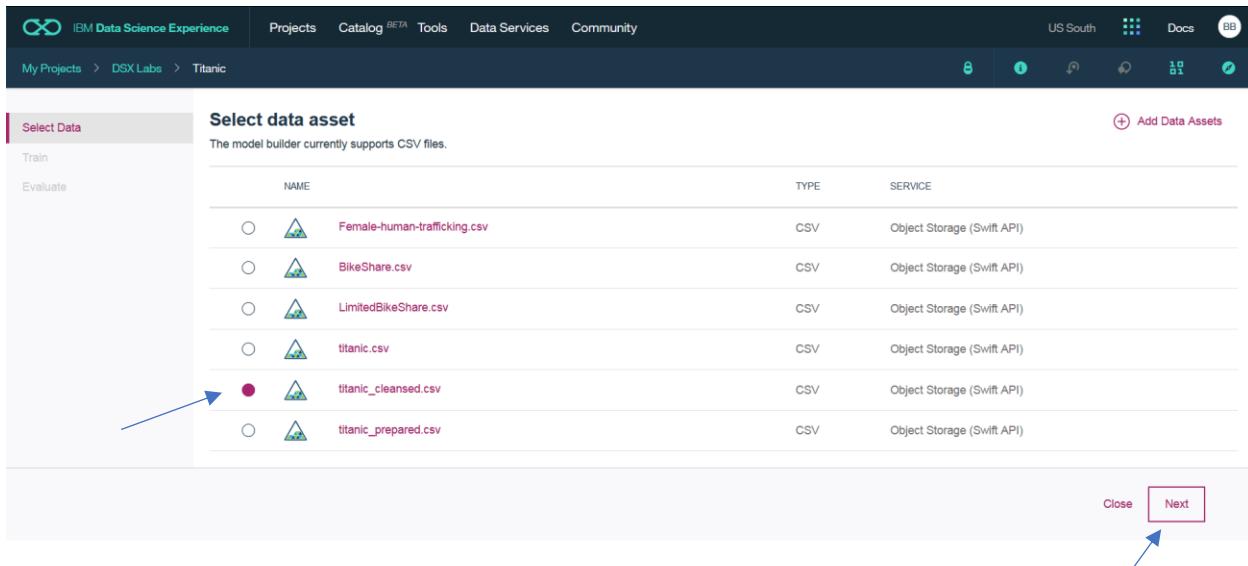
2. Click on **add models**.

The screenshot shows the same IBM Data Science Experience interface as the previous one, but with a different focus. The 'Analytics Assets' tab is still selected. In the main content area, there are three sections: 'Notebooks' (with two items listed), 'Models' (with two items listed), and 'Streaming Pipelines'. A red arrow points to the '+ add models' button located next to the 'Models' section. The bottom of the screen shows a Windows taskbar with various icons.

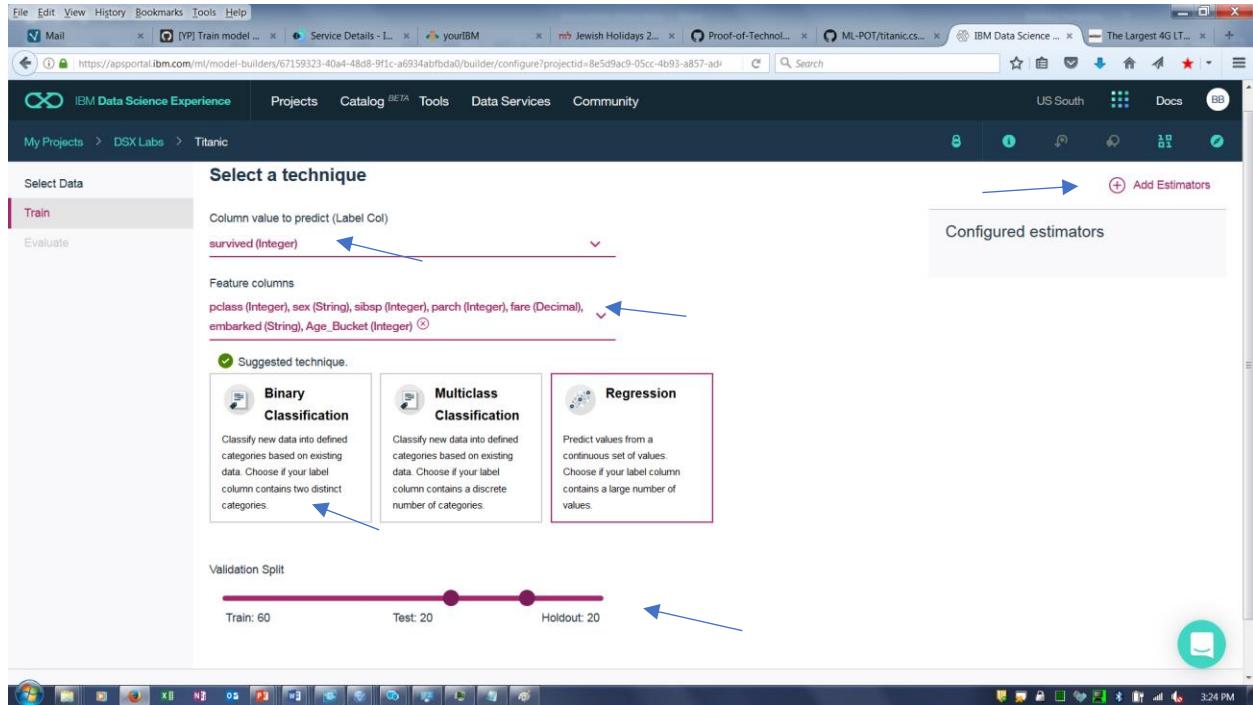
3. Enter the Model **Name**, **Description**, Select **Manual**, and click on **Create**.



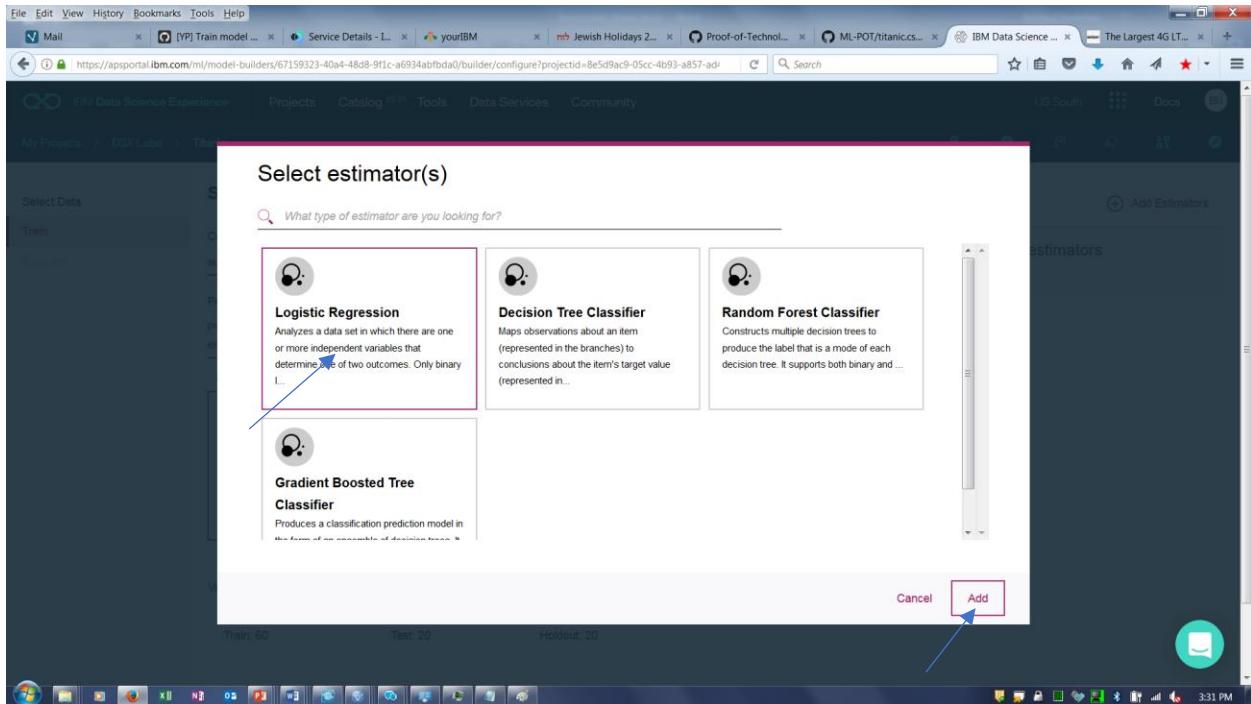
4. Click on the titanic_cleansed.csv and click on **Next**



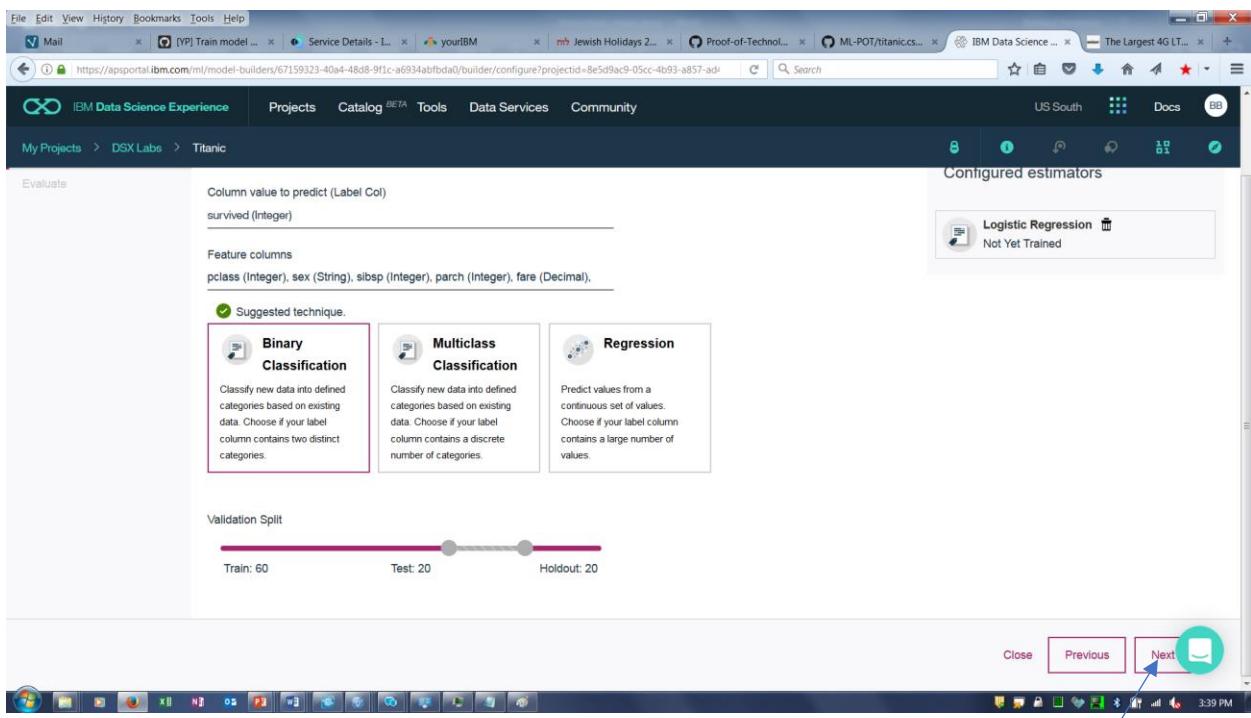
5. For **Column value to predict (Label Col)** select **survivor**. For **Feature columns** select the following features (**pclass, sex, sibsp, parch, fare, embarked, Age_Bucket**) . Click on the **Binary Classification Box** (which is suggested by the service). Adjust the **Validation Split** as desired.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



- 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
●	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	View

Input Schema

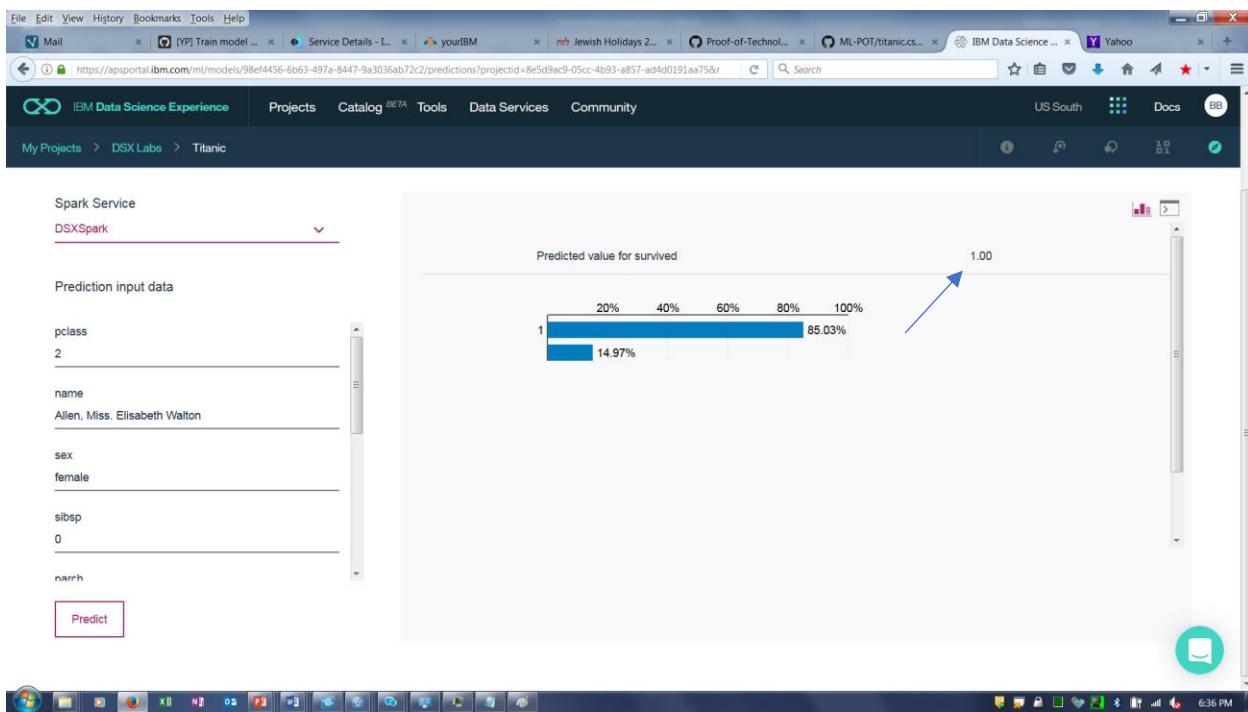
10. Select **Test** and then **Predict**

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 header for 'Spark Service' and '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 in the screenshot below. Another blue arrow points from the 'Predict' button to the predicted result.

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 links 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 refresh icon. Below the title, there are tabs for 'Overview', 'Evaluation', 'Deployments' (which is underlined in red), and 'Test'. Under the 'Deployments' tab, there's a table with columns 'NAME' and 'DEPLOYMENT TYPE'. A note 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 pink header bar with the text 'Deploy model'. Below the header, there are two input fields: 'Deployment Type' set to 'Online' (with a blue arrow pointing to it) and 'Name' set to 'Titanic_Deployment' (with a blue arrow pointing to it). 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 of the screenshot 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 header with 'IBM Data Science Experience' and navigation links for 'Projects', 'Catalog BETA', 'Tools', 'Data Services', and 'Community'. The 'US South' region is selected. A sidebar on the left shows 'My Projects > DSX Labs > Titanic'. The main content area displays deployment details for a successfully created deployment:

- Your deployment was successfully created.**
- Model builder details**: View
- Training data schema**: View
- Input data schema**: View
- Runtime environment**: spark-2.0
- Training date**: 26 Aug 2017, 6:30 PM

Below this, the **Deployments** section lists the deployed service:

NAME	DEPLOYMENT TYPE	ACTIONS
Titanic_Deployment	Online	...

Two blue arrows point to the success message and the deployment entry in the list.

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 'Deployment Details' page for the 'Titanic_Deployment' service. The top navigation bar and sidebar are identical to the previous screenshot. The main content area is titled 'Deployment Details' and includes tabs for 'Details' (selected) and 'Test API'. The 'Test API' tab is highlighted with a blue arrow pointing to it. The deployment details are listed as follows:

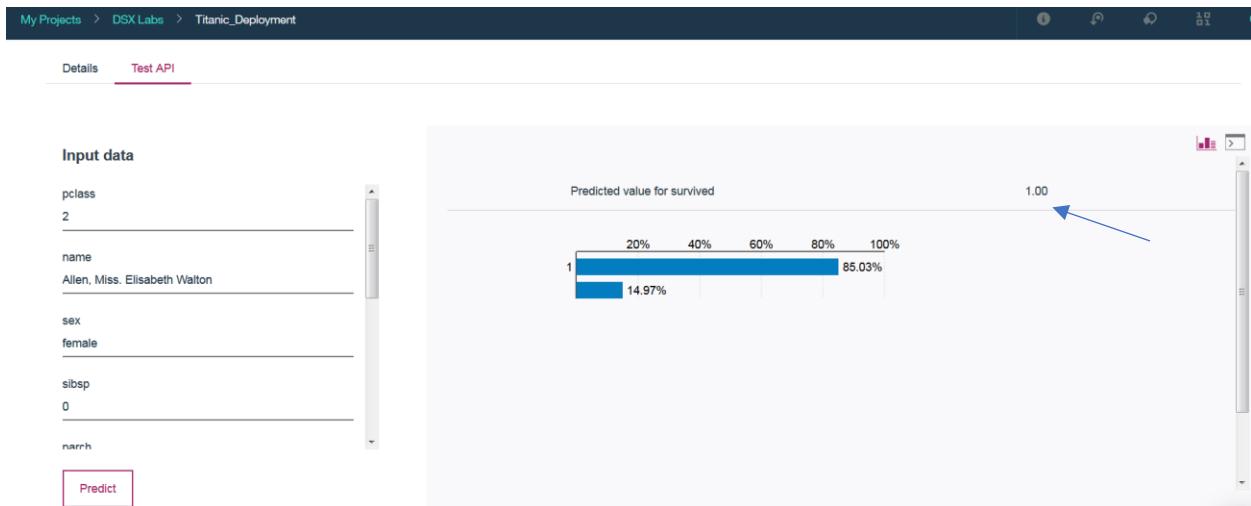
Name	Titanic_Deployment
Type	online
Scoring End Point	https://ibm-watson-ml.mybluemix.net/v3/wml_instances/374817e5-8365-42da-a434-cb20e3d1fba4/published_models/98ef4456-6b63-497a-8447-9a3036ab72c2/deployments/00c07a3c-61ec-45ef-8cdde-213c18f1b448/online
Status	ACTIVE
Associated Model	Titanic

A blue arrow points from the 'Test API' tab to the scoring endpoint URL. The Windows taskbar at the bottom shows various pinned icons and the current time as 6:41 PM.

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.

The screenshot shows the 'Test API' tab of a Watson Studio project. On the left, there's a form titled 'Input data' with fields for pclass (2), name (Allen, Miss. Elisabeth Walton), sex (female), sibsp (0), and parch (0). Below the form is a 'Predict' button. A blue arrow points from the 'Predict' button towards the right side of the screen.

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.

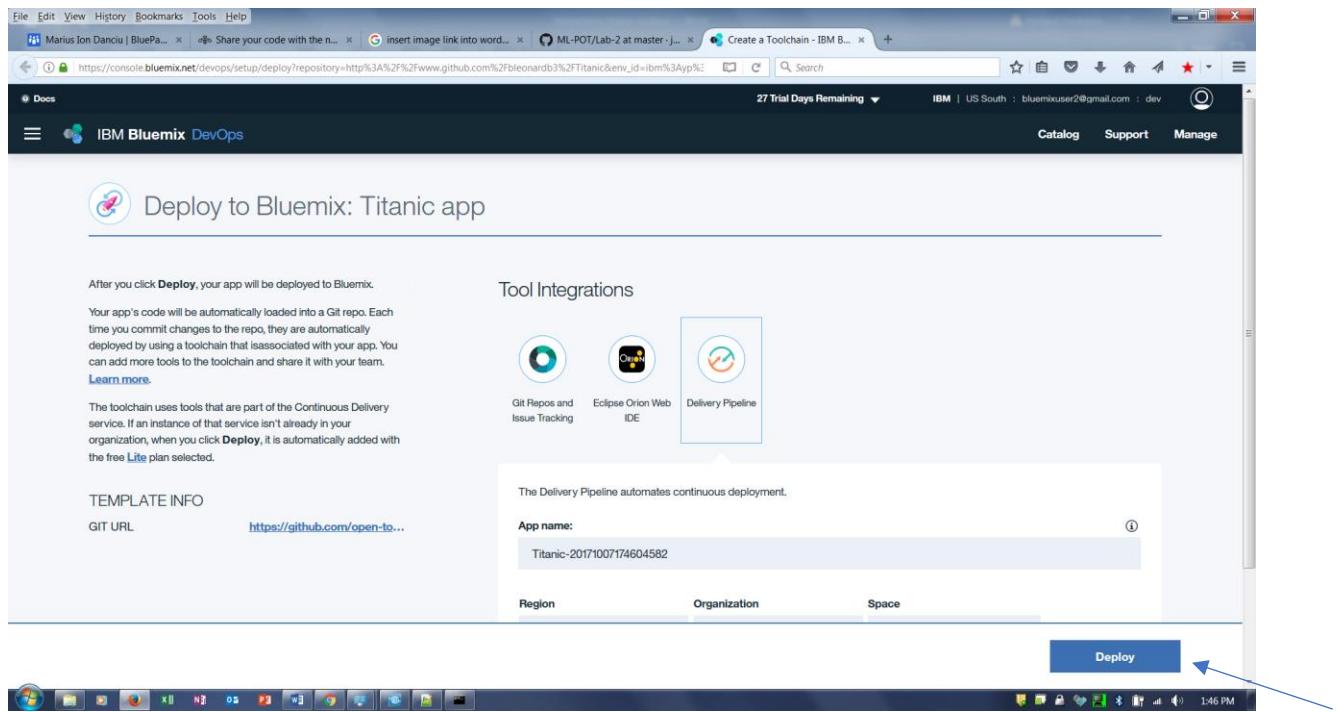
The toolchain uses tools that are part of the Continuous Delivery service. If an instance of that service isn't already in your organization, when you click **Deploy**, it is automatically added with the free [Lite](#) plan selected.

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. Click on the **Deploy** button.

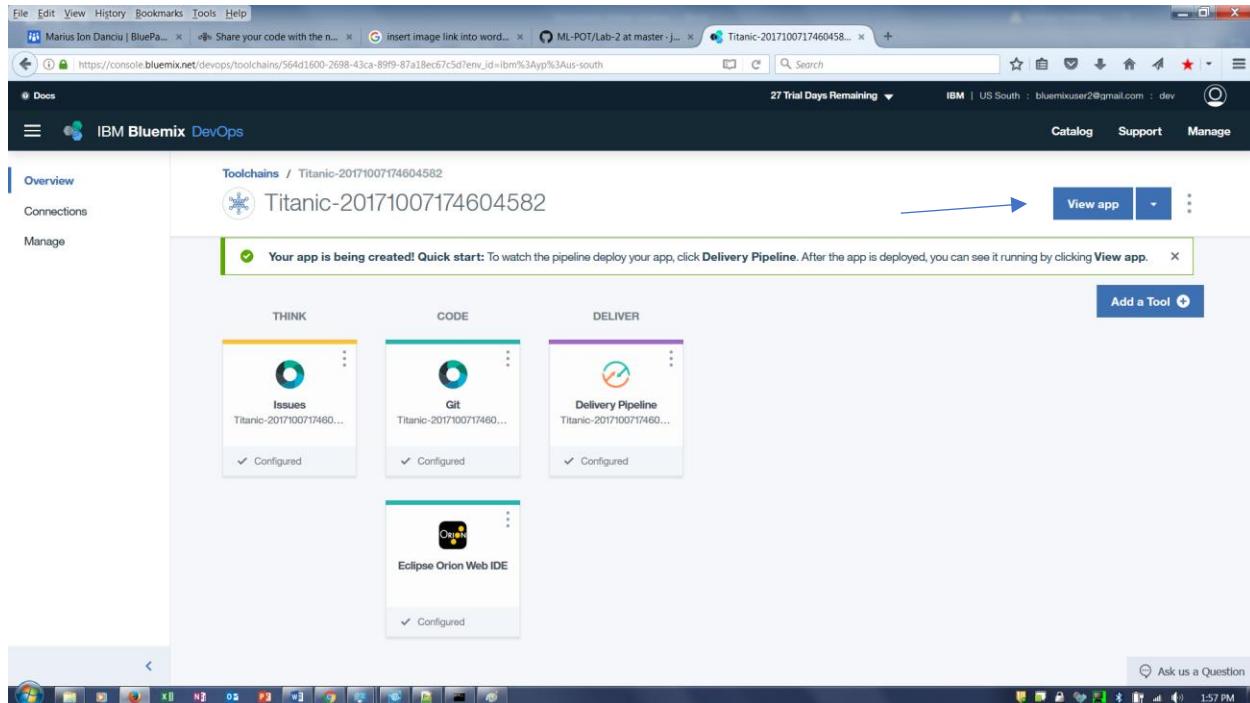


3. Your app is being created! To watch the pipeline deploy your app, click **Delivery Pipeline**.

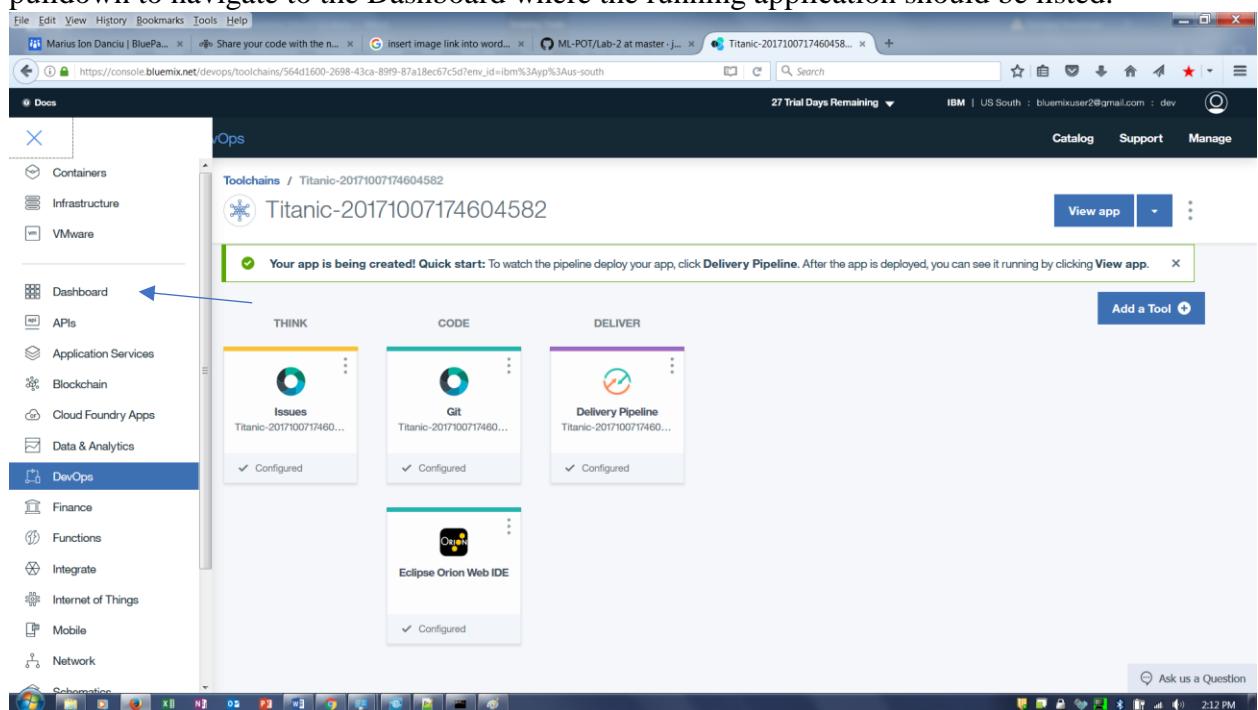
- After the app is deployed successfully (should say Deployed passed now in the Deploy stage), return to the Delivery Pipeline by clicking on the vertical ellipse and click on View Toolchain.

The screenshot shows the 'Delivery Pipeline' page for the app 'Titanic-20171007174604582'. The page is divided into two main sections: 'Build Stage' and 'Deploy Stage'. Both sections show a 'STAGE PASSED' status. The 'Build Stage' section includes details about the last commit, jobs, and execution results. The 'Deploy Stage' section includes details about the last input, jobs, and execution result. At the top right, there is a blue button labeled 'Add Stage' with a plus sign. On the far right, there is a vertical ellipsis icon (three dots) with a blue arrow pointing to it.

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.



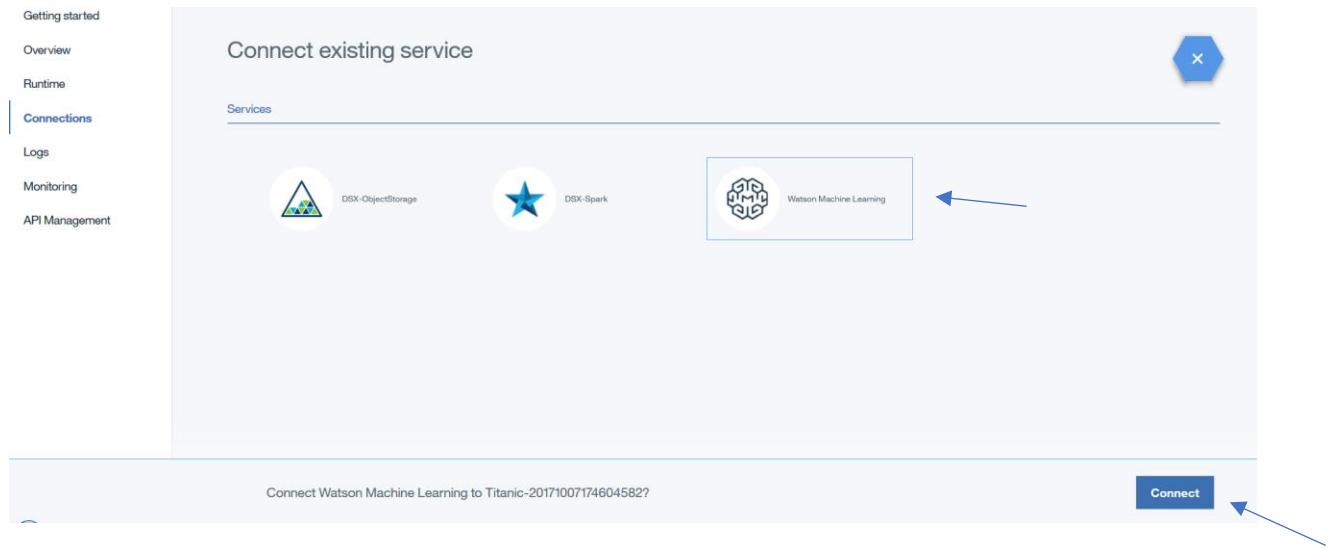
7. We are now going to connect the application to the Watson Machine Learning service that was created earlier. Click on the application name.

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.

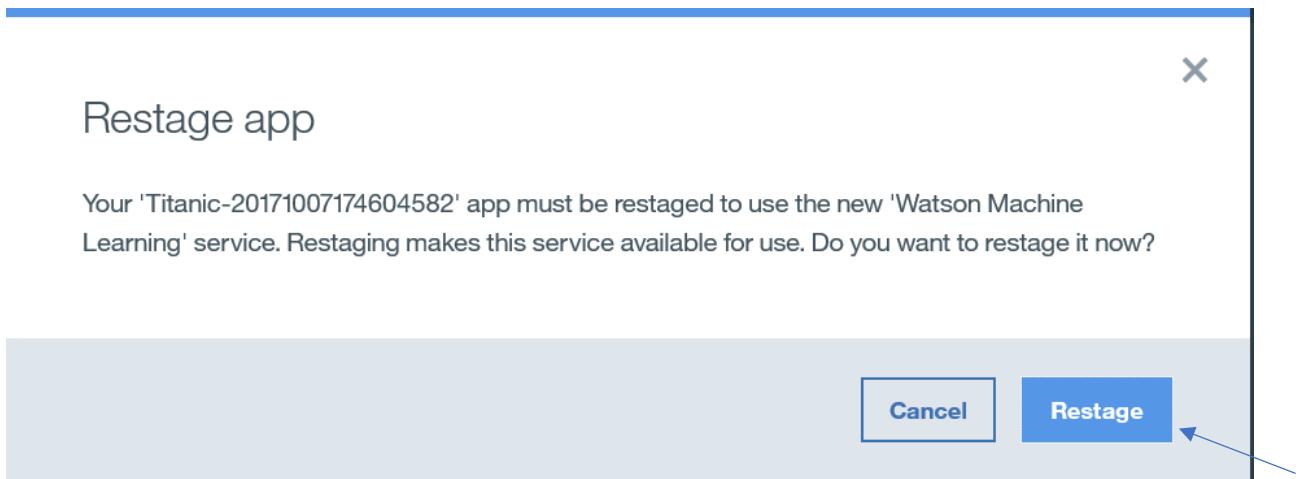
8. Scroll down until you see the Connections panel. Click on Connect Existing.

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 shows 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 current and estimated charges, with a note that current and estimated cost excludes connected services and a 'View full usage details' button.

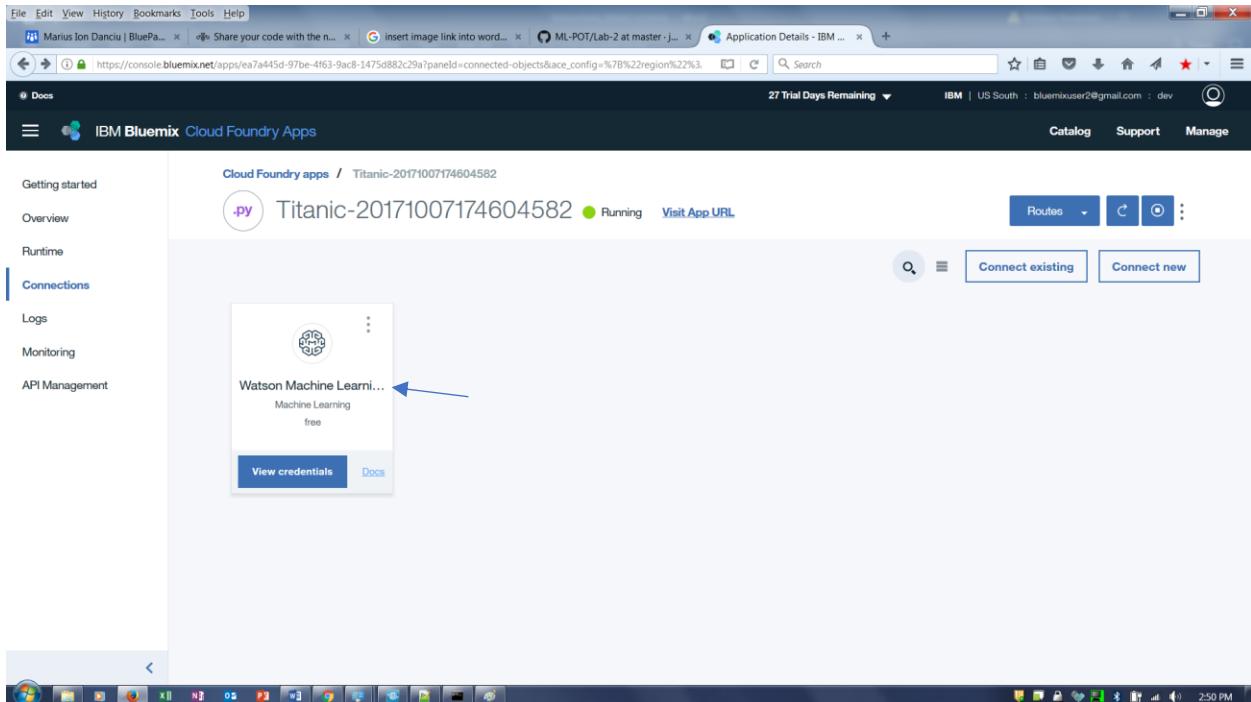
9. You should see at least 3 services listed. Two services are automatically provisioned for the Data Science Experience (Spark service, and Object Storage service). Click on the **Watson Machine Learning** service, and then click on **Connect**.



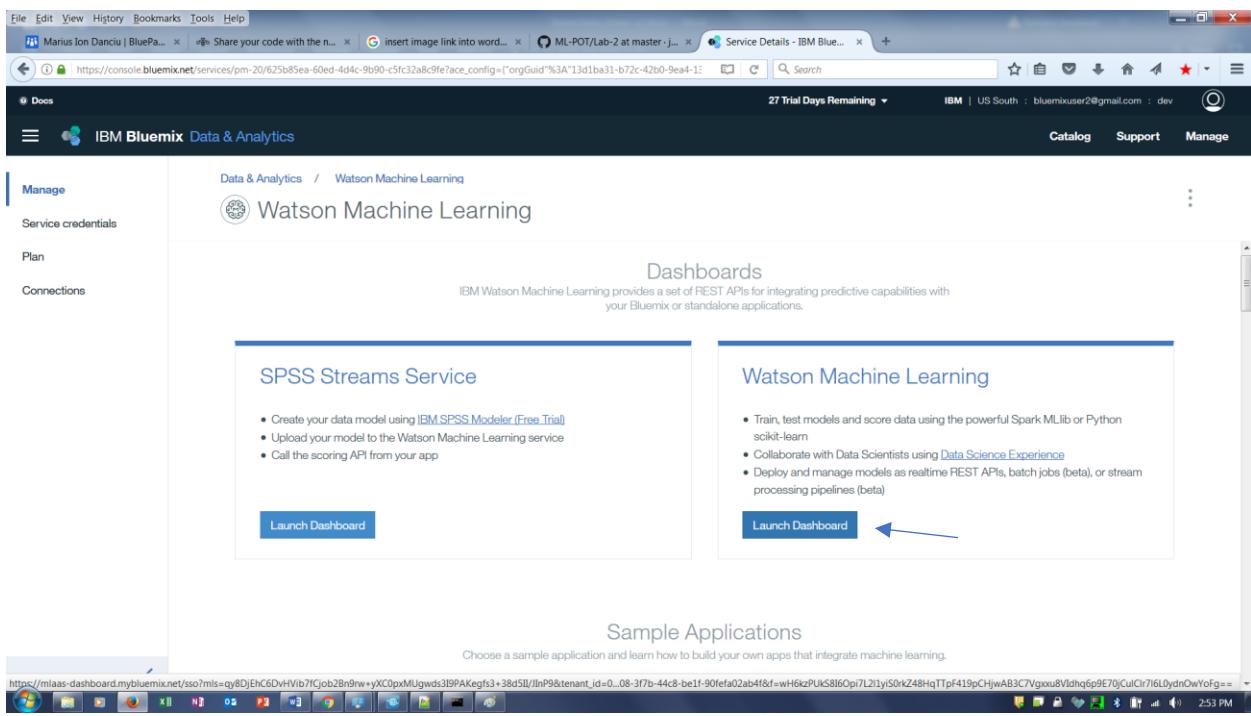
10. You will get a pop up that asks to Restage the application. Click on **Restage**.



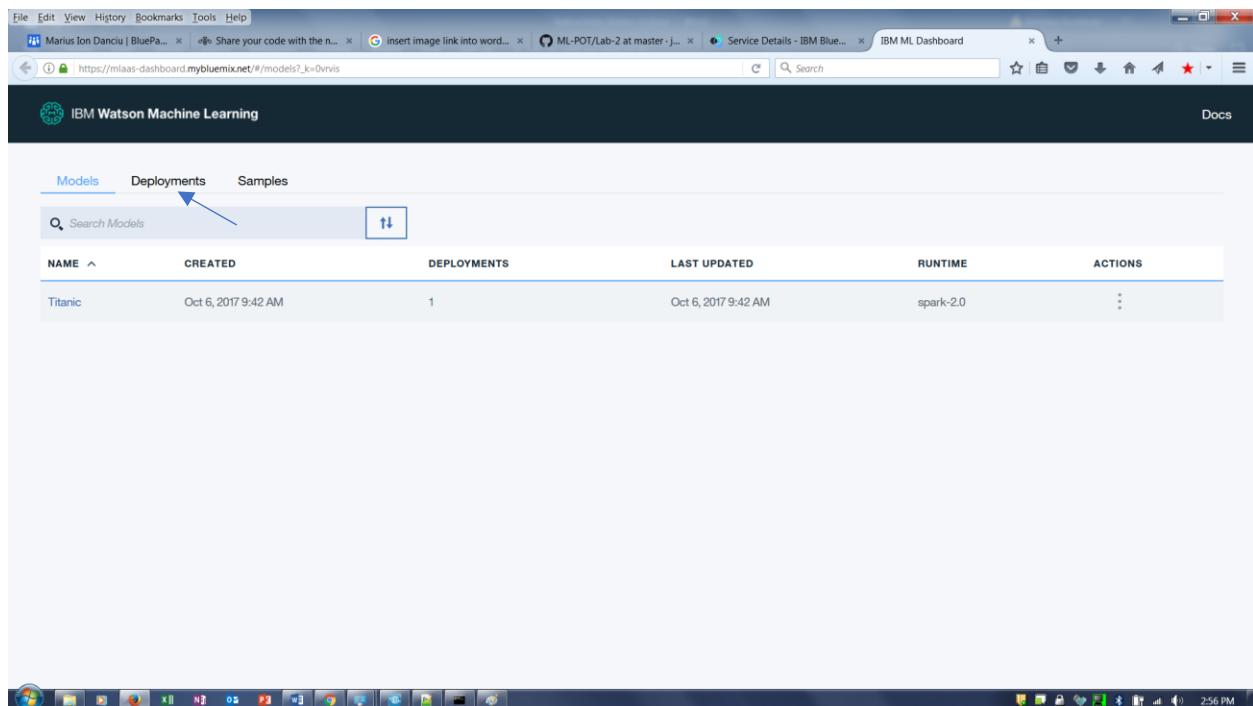
11. We now need to copy the scoring endpoint to the clipboard. Click on the **Watson Machine Learning** service.



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



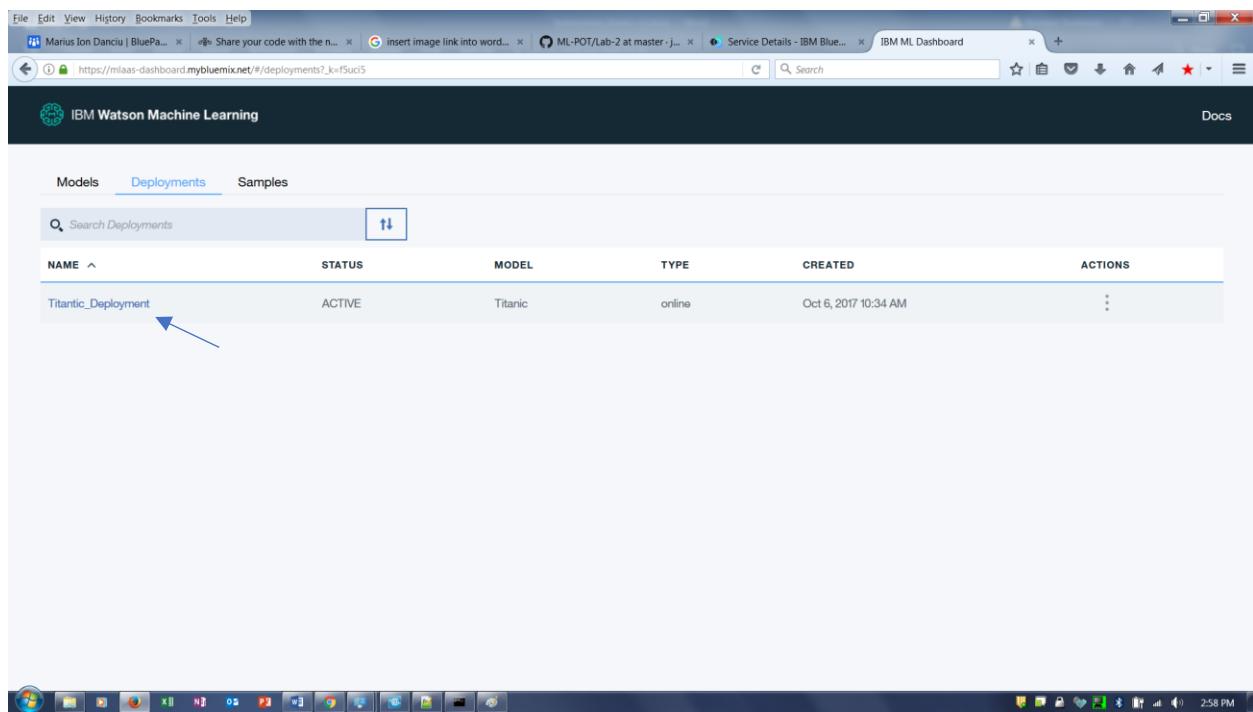
13. Click on the **Deployment** tab.



The screenshot shows the IBM Watson Machine Learning dashboard. The 'Models' tab is currently selected. A blue arrow points from the text '14. Click on the Titanic Deployment.' to the 'Deployments' tab. The dashboard displays a table with one row for the 'Titanic' model, showing details like creation date, number of deployments, and runtime.

NAME	CREATED	DEPLOYMENTS	LAST UPDATED	RUNTIME	ACTIONS
Titanic	Oct 6, 2017 9:42 AM	1	Oct 6, 2017 9:42 AM	spark-2.0	⋮

14. Click on the Titanic Deployment.



The screenshot shows the IBM Watson Machine Learning dashboard with the 'Deployments' tab selected. A blue arrow points from the text '15. Scroll down to API details and copy the Scoring Endpoint value “http.....etc”' to the 'Titanic_Deployment' row in the table. The table lists the deployment status, model used, type, and creation date.

NAME	STATUS	MODEL	TYPE	CREATED	ACTIONS
Titanic_Deployment	ACTIVE	Titanic	online	Oct 6, 2017 10:34 AM	⋮

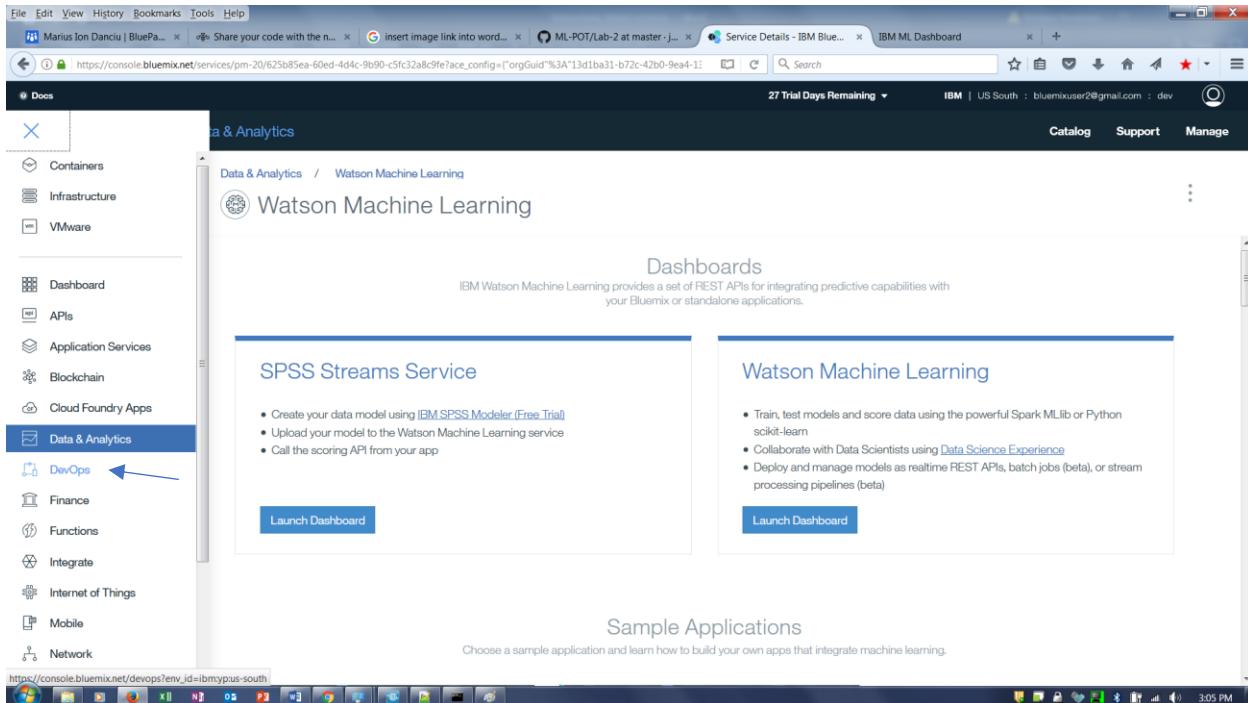
15. Scroll down to API details and copy the Scoring Endpoint value “http.....etc”

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

API Details

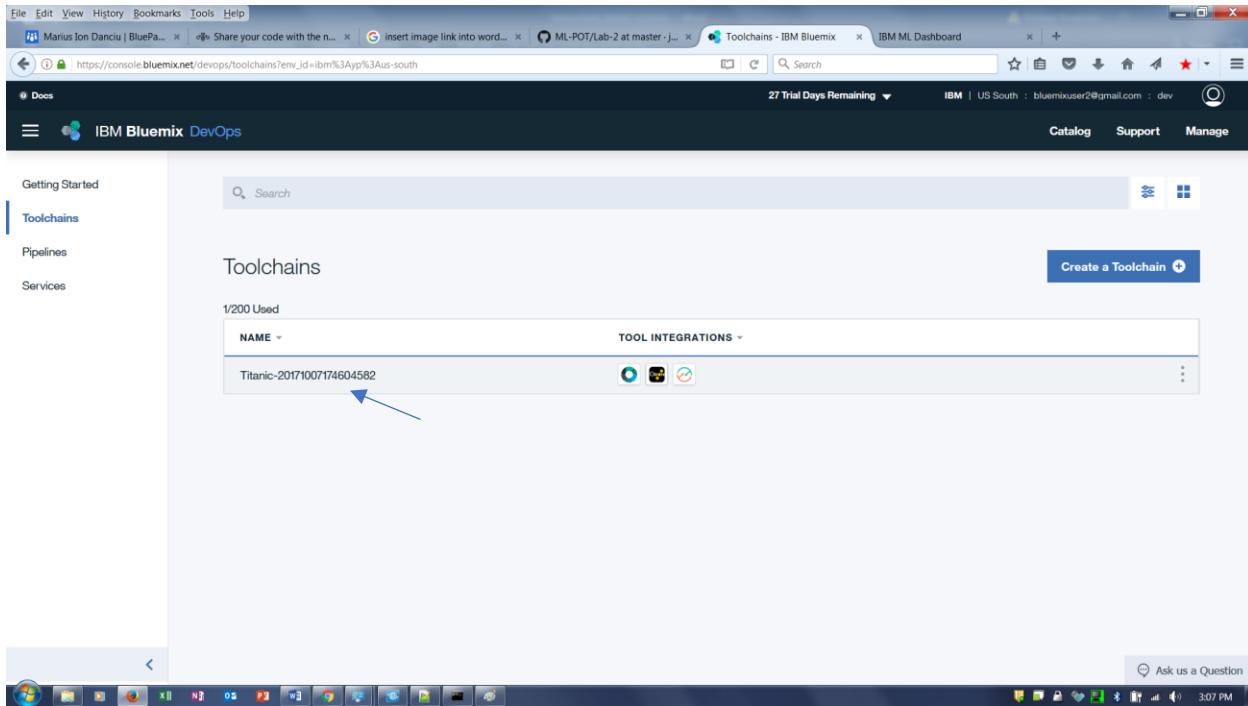
Scoring Endpoint: https://ibm-watson-ml.mybluemix.net/v3/wml_instances/625b85ea-60ed-4d4c-9b90-c5fc32a8c9fe/published_models/0788b520-0f19-4f7c-9ee5-03a9f7ed899d

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.

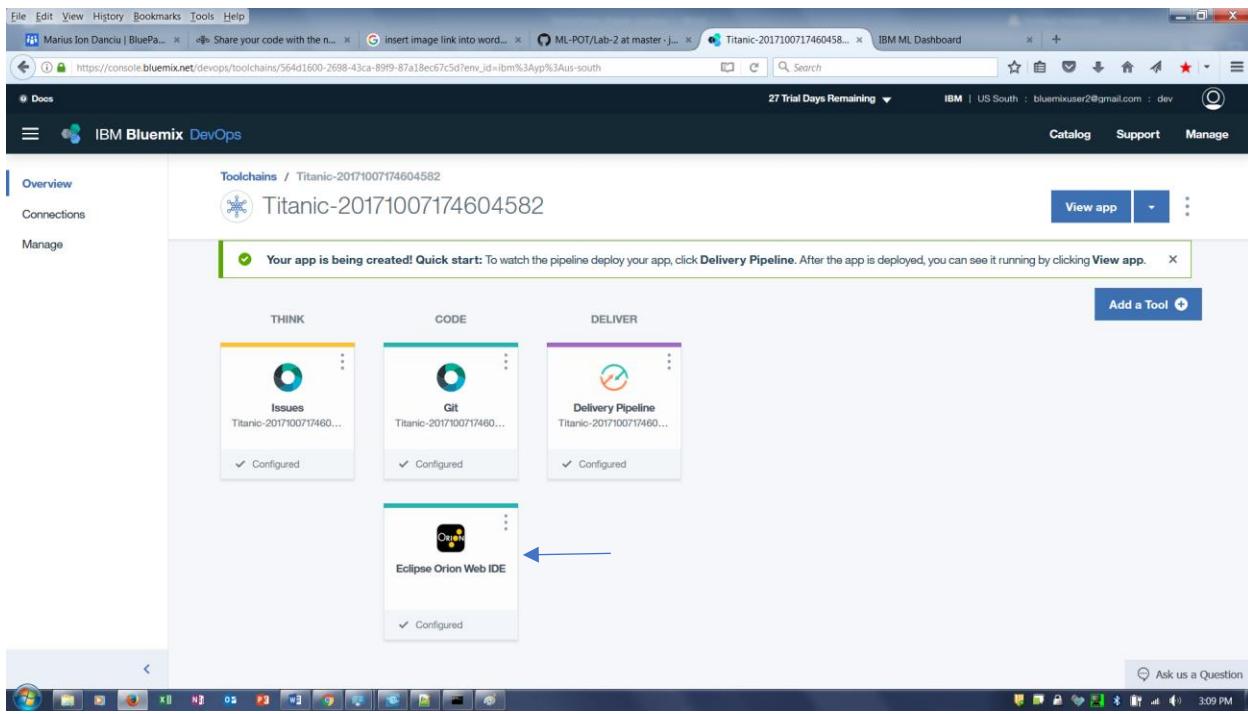


The screenshot shows the IBM ML Dashboard interface. At the top, there are several tabs: 'Service Details - IBM Blue...', 'ML-POT/Lab-2 at master', 'IBM ML Dashboard', and others. Below the tabs, there's a search bar and navigation icons. The main content area has sections for 'Dashboards', 'SPSS Streams Service', and 'Watson Machine Learning'. In the bottom left corner, there's a sidebar with various service categories like 'Containers', 'Infrastructure', 'VMware', 'Dashboard', 'APIs', 'Application Services', 'Blockchain', 'Cloud Foundry Apps', 'Data & Analytics', 'DevOps' (which is highlighted with a blue arrow), 'Finance', 'Functions', 'Integrate', 'Internet of Things', 'Mobile', and 'Network'. The 'DevOps' link is underlined and has a blue arrow pointing to it.

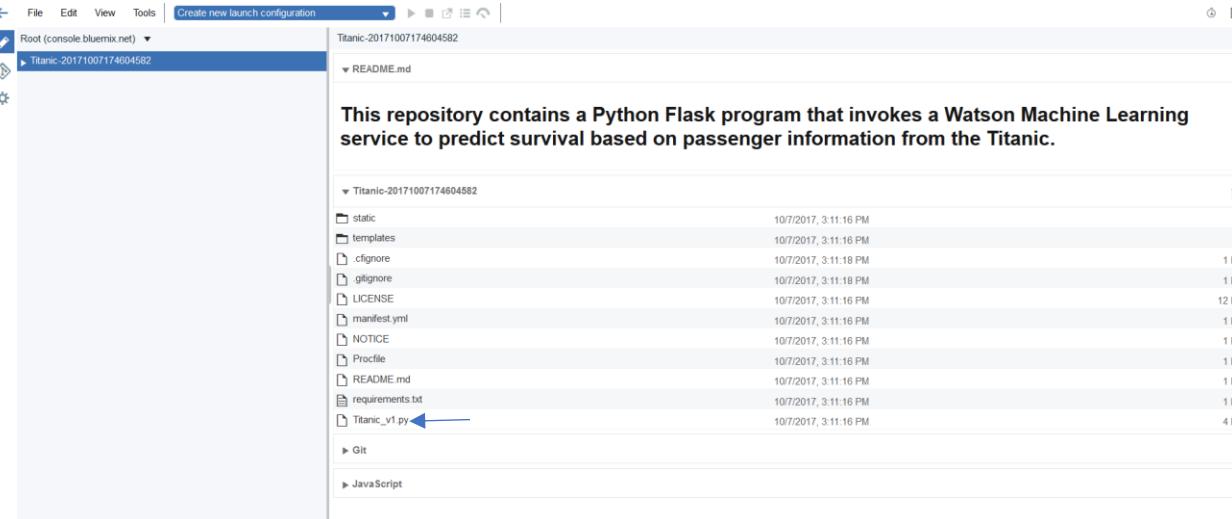
17. We are now going to paste the scoring endpoint into the application code. Click on the Toolchain.



18. Click on the Eclipse Orion Web IDE.



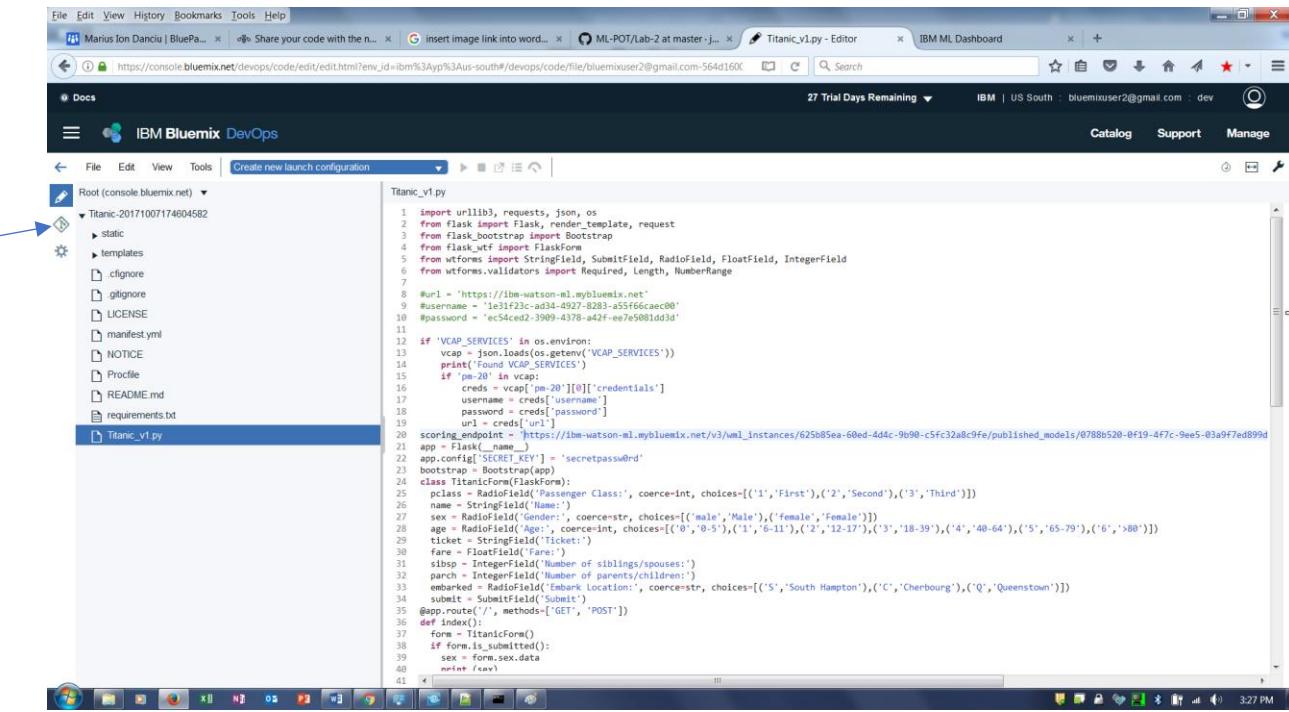
19. Click on the Titanic_v1 file.



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

- static
- templates
- cignore
- gitignore
- LICENSE
- manifest.yml
- NOTICE
- Profile
- README.md
- requirements.txt
- Titanic_v1.py**

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.

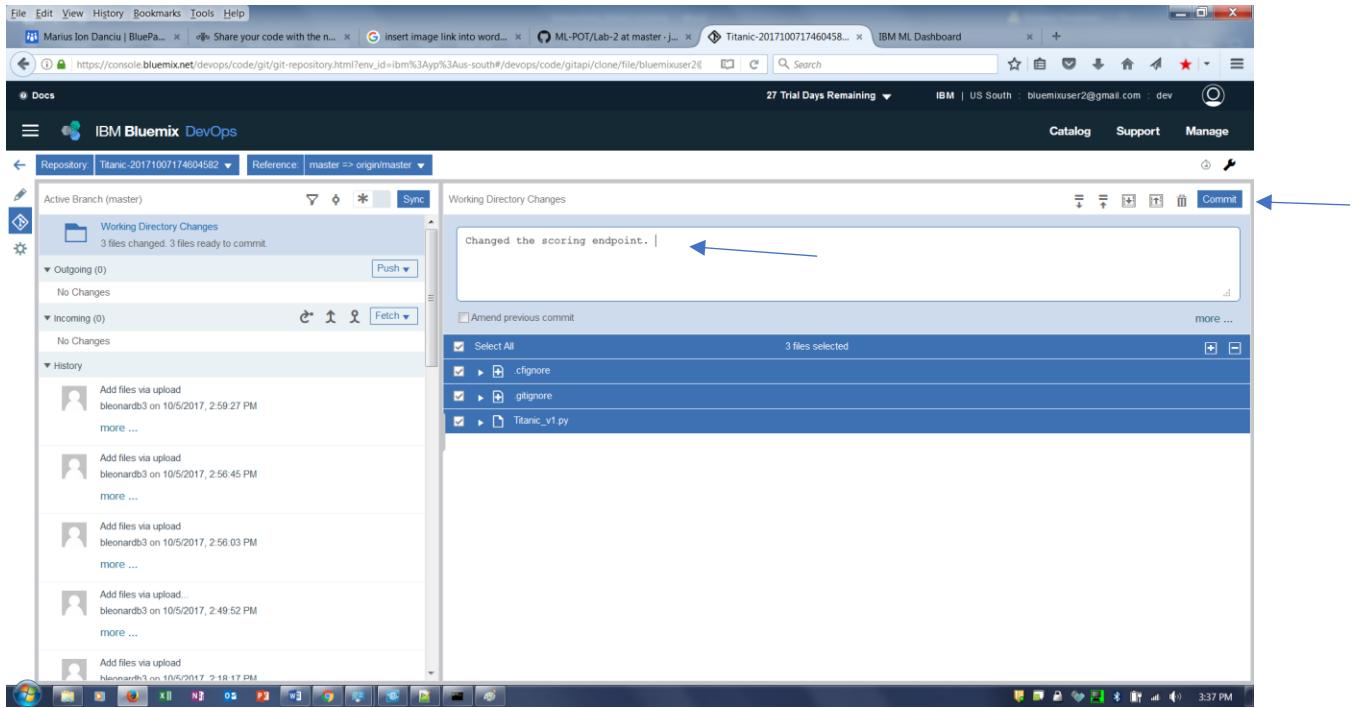


```

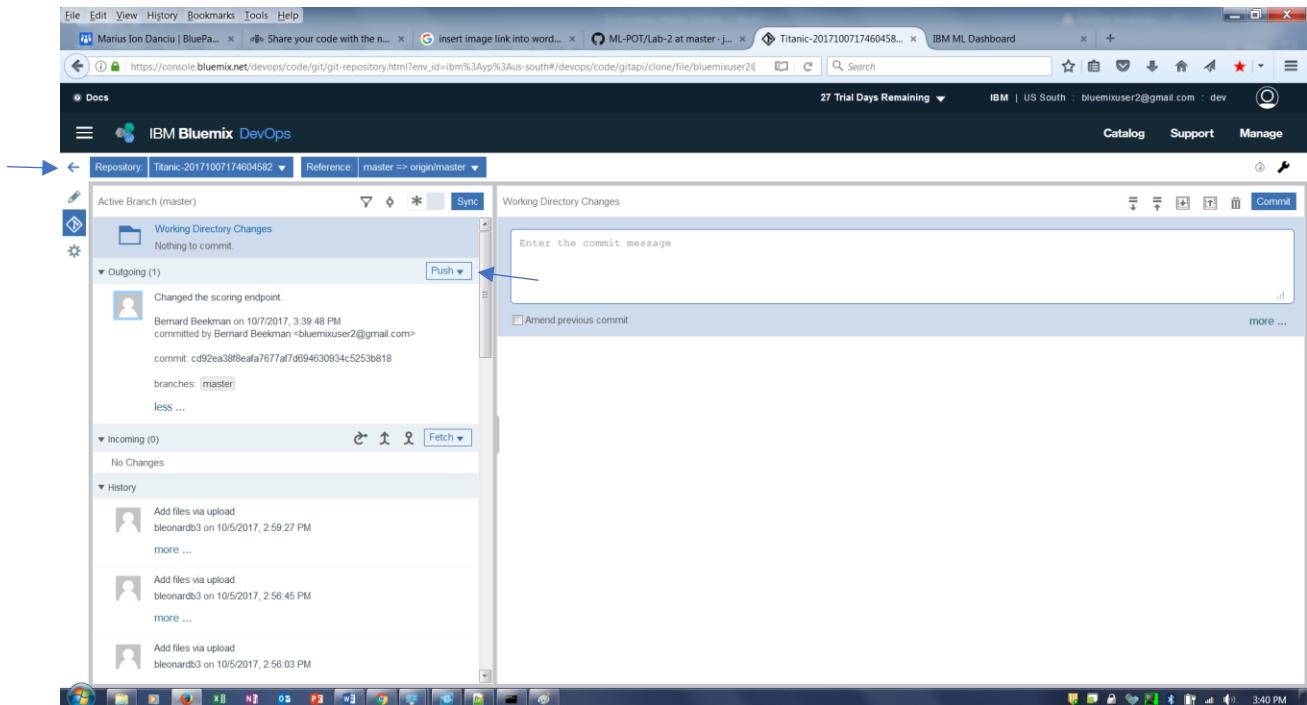
1 import urllib3, requests, json, os
2 app = Flask(__name__)
3 from flask_bootstrap import Bootstrap
4 from flask_wtf import FlaskForm
5 from wtforms import StringField, SubmitField, RadioField, FloatField, IntegerField
6 from wtforms.validators import Required, Length, NumberRange
7
8 #url = 'https://ibm-watson-ml.mybluemix.net'
9 #username = '1e1f23c-ad34-4927-8283-a55f66caec00'
10 #password = 'ec34cd2-3909-4378-a2cf-ee7e59b1dd3d'
11
12 if 'VCAP_SERVICES' in os.environ:
13     vcap = json.loads(os.getenv('VCAP_SERVICES'))
14     print('Found VCAP_SERVICES')
15     if 'wml': # WML VCAP entry
16         creds = vcap['wml'][0]['credentials']
17         username = creds['username']
18         password = creds['password']
19         url = creds['url']
20     scoring_endpoint = url + 'v3/wml_instances/625b85ea-60ed-4d4c-9b90-c5fc32a8c9fe/published_models/0788b520-0f19-4f7c-9ee5-03a9f7ed899d'
21     app.config['SECRET_KEY'] = 'secretpassw0rd'
22 bootstrap = Bootstrap(app)
23 bootstrap = Bootstrap(app)
24 class TitanicForm(FlaskForm):
25     name = StringField('Name:')
26     sex = RadioField('Sex', choices=[('male','Male'),('female','Female')])
27     age = RadioField('Age', coerce=int, choices=[('0-5'),('1-6-11'),('2-12-17'),('3-18-39'),('4-40-64'),('5-65-79'),('6+', '>80')])
28     ticket = StringField('Ticket:')
29     fare = FloatField('Fare:')
30     sibsp = IntegerField('Number of siblings/spouses:')
31     parch = IntegerField('Number of parents/children:')
32     embarked = RadioField('Embarked location:', coerce=str, choices=[('S','South Hampton'),('C','Cherbourg'),('Q','Queenstown')])
33     submit = SubmitField('Submit')
34     @app.route('/', methods=['GET', 'POST'])
35     def index():
36         form = TitanicForm()
37         if form.is_submitted():
38             sex = form.sex.data
39             sex = sex
40
41

```

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!](#)