

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 IBM Cloud dashboard:

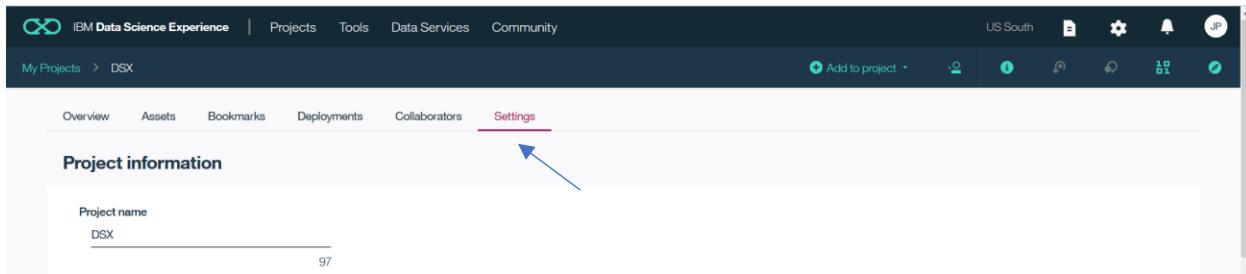
- Watson Machine Learning
- Object Storage
- Apache Spark

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



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

Associated Services			
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.

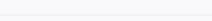
Pricing Plan: Monthly Process shown above reflect the: **United States**

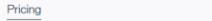
Plan	Features	Pricing
Free	Service instance (5 models per instance) 5,000 predictions 5 compute hours	Free

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

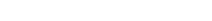














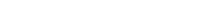




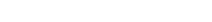


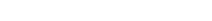













<img alt="A blue arrow points from the 'Machine Learning' section

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 listed, showing 1845 lines and 75.6 KB. The file content is displayed as a table with columns: pclass, survived, name, sex, sibsp, parch, ticket, fare, embarked. A blue arrow points to the 'Raw' button at the top of the table preview.

pclass	survived	name	sex	sibsp	parch	ticket	fare	embarked
1	1	Allan, Miss. Elisabeth Walton	female	0	0	24160	211337500	S
1	1	Allison, Master. Hudson Trevor	male	1	2	13781	151550000	S
1	0	Allison, Miss. Helen Loraine	female	1	2	13781	151550000	S
1	0	Allison, Mr. Hudson Joshua Creighton	male	1	2	13781	151550000	S
1	n	Allison, Mr. Hudson Joshua Creighton	female	1	2	13781	151550000	S

3. Go back to your DSX project. Click on New data asset or the icon.

A screenshot of the IBM Data Science Experience (DSX) interface. The 'Assets' tab is selected. At the top right, there is a '+ New data asset' button and an icon with a plus sign and a document. Below the tabs, there is a search bar and a section for 'Data assets'. A table lists one asset: 'DB2 Warehouse-DSX' (Connection type, Db2 Warehouse on Cloud service, last modified 1 Nov 2017). A blue arrow points to the '+ New data asset' button, and another blue arrow points to the icon in the top right.

4. In the **Files** 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**.

The screenshot shows the DSX interface with the 'Assets' tab selected. The main panel displays a table of data assets, including 'DB2 Warehouse-DSX'. The sidebar on the right has a 'Files' tab open, showing a 'Drop file here or browse' area and a 'Find in storage' search bar. A blue arrow points from the search bar in the main panel to the 'Drop file here or browse' area in the sidebar.

Step 3: Create a Model to predict survival

1. Click on the Assets Tab

The screenshot shows the DSX interface with the 'Assets' tab selected. The main panel displays tables for Data assets, Notebooks, and Models. In the Data assets section, there is a CSV file named 'titanic_cleaned.csv' and a connection named 'DB2 Warehouse-DSX'. The sidebar on the right has a 'Files' tab open, showing a 'Drop file here or browse' area and a 'Find in storage' search bar. A blue arrow points from the search bar in the main panel to the 'Drop file here or browse' area in the sidebar.

2. Click on New model.

IBM Data Science Experience | Projects Tools Data Services Community

My Projects > DSX

Overview Assets Bookmarks Deployments Collaborators Settings

What assets are you looking for?

Data assets

NAME	TYPE	SERVICE	LAST MODIFIED	ACTIONS
titanic_cleaned.csv	CSV	Object Storage (Swift API)		⋮
DB2 Warehouse-DSX	Connection	Db2 Warehouse on Cloud	1 Nov 2017	⋮

Notebooks

NAME	SHARED	SCHEDULED	STATUS	LANGUAGE	LAST EDITOR	LAST MODIFIED	ACTIONS
Connect and Interact with DB2 Warehouse			○	Python 3.5	Joel Patterson	1 Nov 2017	✍️ ⋮

Models

NAME	STATUS	RUNTIME	LAST MODIFIED	ACTIONS
you currently have no models				

Drop file here or browse your files to add a new file

Find in storage

titanic_cleaned.csv

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

IBM Data Science Experience | Projects Catalog BETA Tools Data Services Community

US South Docs BB

Create new model BETA

Name
Titanic

Description
Machine Learning Model for the Titanic Data Set

Machine Learning Service
Watson Machine Learning

Spark Service
DSXSpark

Automatic
Prepare my data and create a model automatically

Manual
Let me prepare my data and select which models to train

Need something more flexible? Create a [notebook](#) or design a [flow](#).

Cancel Create

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

Select data asset

The model builder currently supports CSV files.

NAME	TYPE	SERVICE
titanic_cleansed.csv	CSV	Object Storage (Swift API)

Add Data Assets

Close Next

- 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.

Select a technique

Column value to predict (Label Col)
survived (Integer)

Feature columns
pclass (Integer), sex (String), sibsp (Integer), parch (Integer), fare (Decimal), embarked (String), Age_Bucket (Integer) ⚡

Suggested technique.

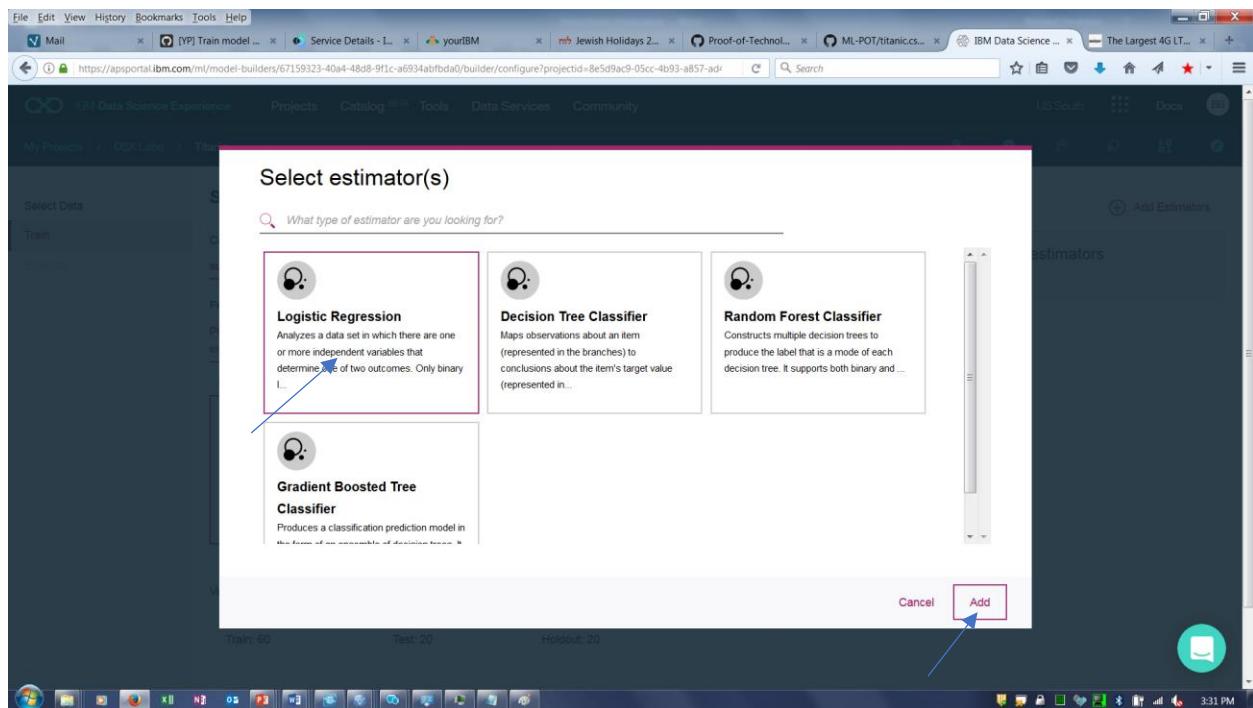
Binary Classification Classify new data into defined categories based on existing data. Choose if your label column contains two distinct categories.	Multiclass Classification Classify new data into defined categories based on existing data. Choose if your label column contains a discrete number of categories.	Regression Predict values from a continuous set of values. Choose if your label column contains a large number of values.
---	---	---

Validation Split
Train: 60 Test: 20 Holdout: 20

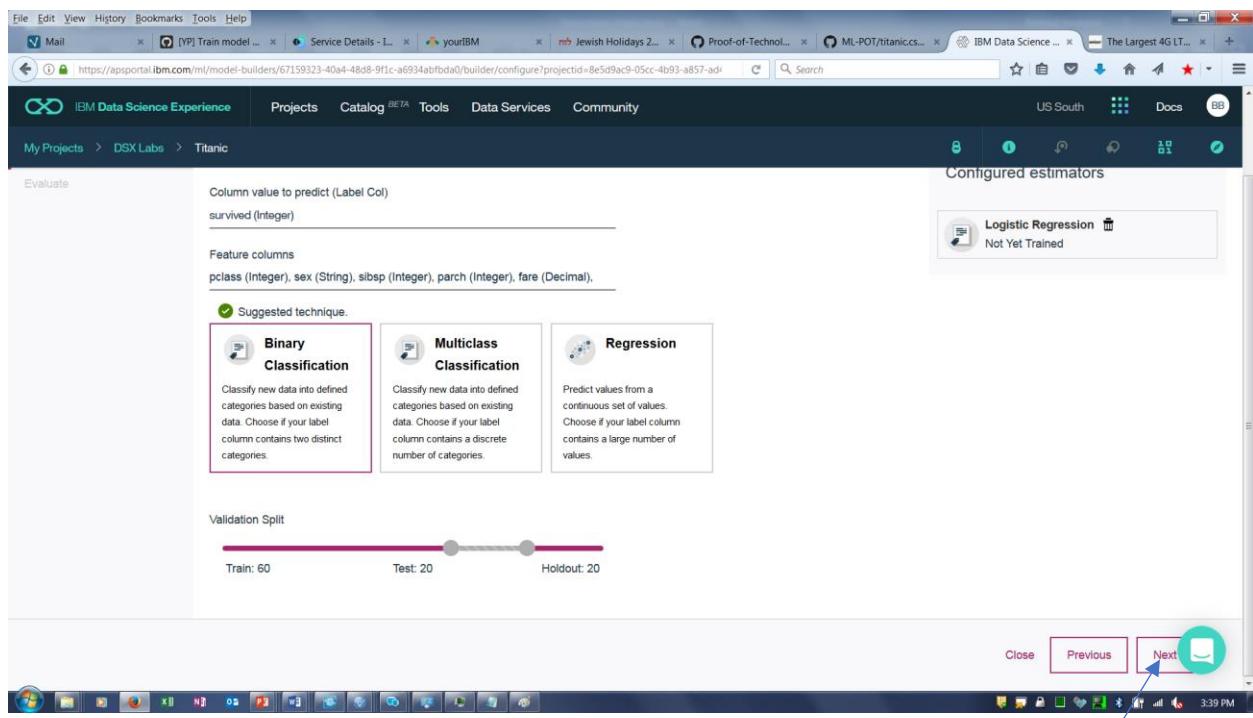
Add Estimators

Configured estimators

- Select all models. You can select more if you wish to see the results of multiple models (I usually select all of them). 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 the best performing model and then click Save.

Select model

ESTIMATOR TYPE	STATUS	PERFORMANCE	AREA UNDER ROC CURVE	AREA UNDER PR CURVE	LAST EVALUATION	ACTIONS
RandomForestClassifier	Trained & Evaluated	Good	0.83062	0.81338	1 Nov 2017, 1:19 PM	...
GBTClassifier	Trained & Evaluated	Fair	0.75036	0.76747	1 Nov 2017, 1:19 PM	...
DecisionTreeClassifier	Trained & Evaluated	Poor	0.57547	0.62503	1 Nov 2017, 1:18 PM	...
LogisticRegression	Evaluation Error

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

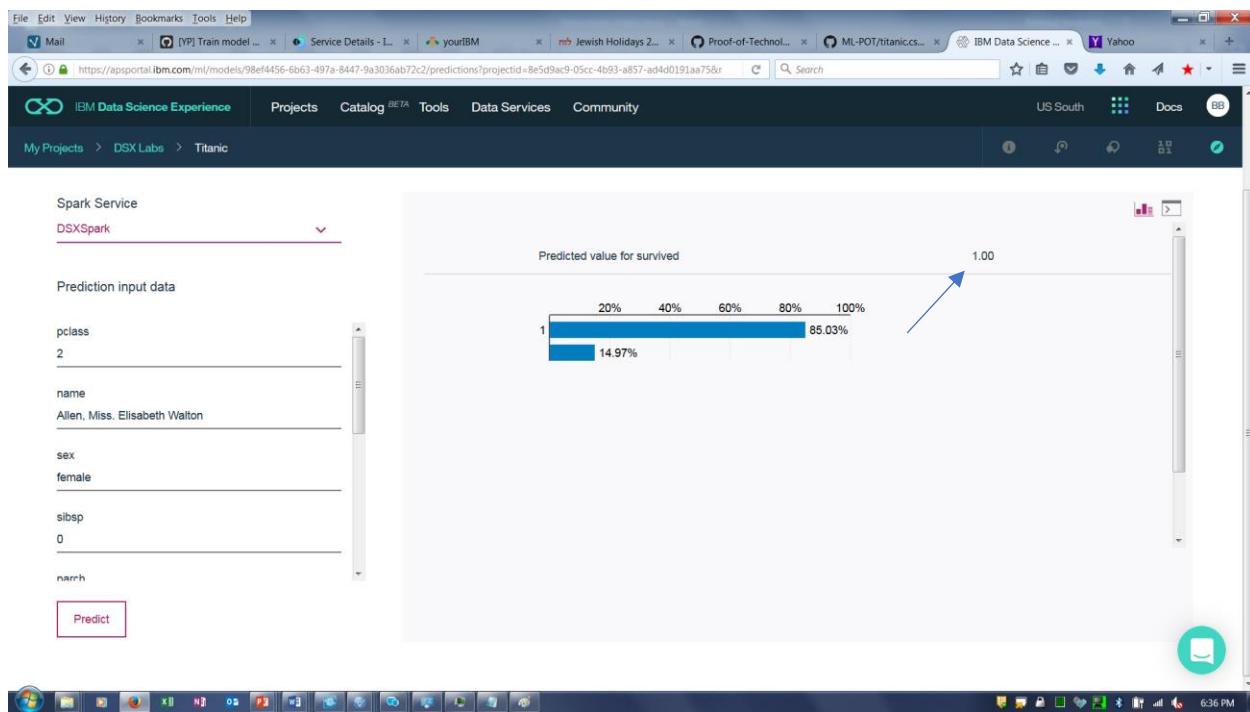
10. Select **Test** and then **Predict**

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 '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 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 icons for 'US South', a user profile, settings, notifications, and a help icon. Below the header, the path 'My Projects > DSX > Titanic' is visible. The main content area is titled 'Titanic' with a subtitle 'Your model is not deployed.' Below this, there are tabs for 'Overview', 'Evaluation', 'Deployments' (which is highlighted in red), and 'Test'. A table with columns 'NAME', 'DEPLOYMENT TYPE', and 'ACTIONS' follows. In the top right corner of the table area, there's a blue 'Add Deployment' button with a plus sign, which is also highlighted with a blue arrow.

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 being highlighted by a red box and a blue arrow pointing to it. The background shows the main interface with various tabs like 'Label column', 'Model builder details', etc., and a sidebar with a 'Deployments' section. The system status bar at the bottom indicates '6:38 PM'.

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

Your deployment was successfully created.

Deployments

NAME	DEPLOYMENT TYPE	ACTIONS
Titanic_Deployment	Online	...

- 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.

Deployment Details

Name	Titanic_Deployment
Type	online
Scoring End Point	https://ibm-watson-ml.mybluemix.net/v3/wml_instances/374817e5-8365-42da-a434-cb20e3d1fba4/published_models/98ef4456-6b53-497a-8447-9a3036ab72c2/deployments/00c07a3c-61ec-45ef-8cde-213c18f1b448/online
Status	ACTIVE
Associated Model	Titanic

- 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

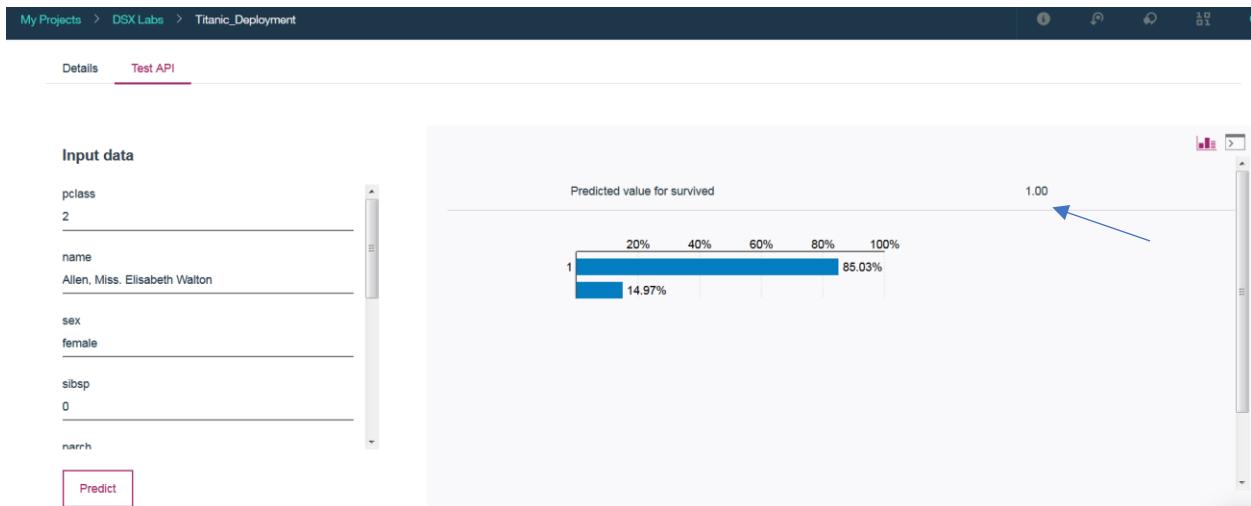
Details Test API

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 IBM Cloud 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.

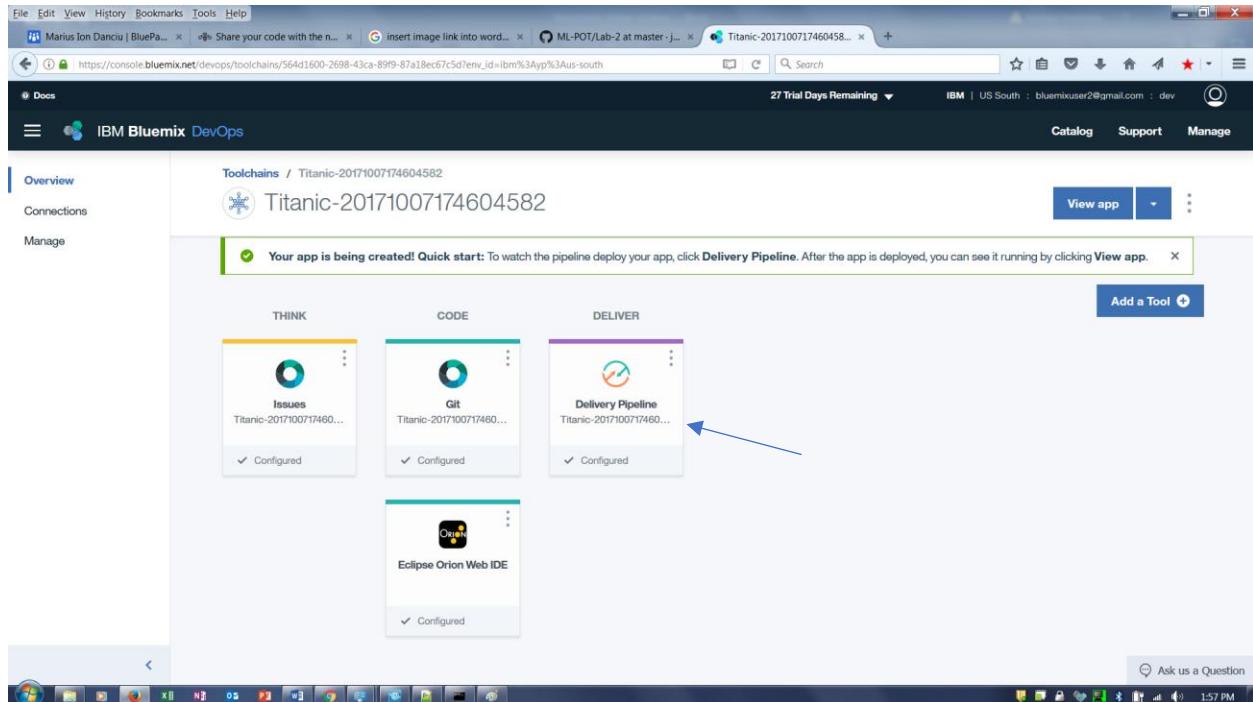
1. Click on the following to deploy a sample Python Flash web application into your IBM Cloud account. Note you may get a message to sign in to IBM Cloud.

Deploy to IBM Cloud

2. Click on the **Deploy** button.

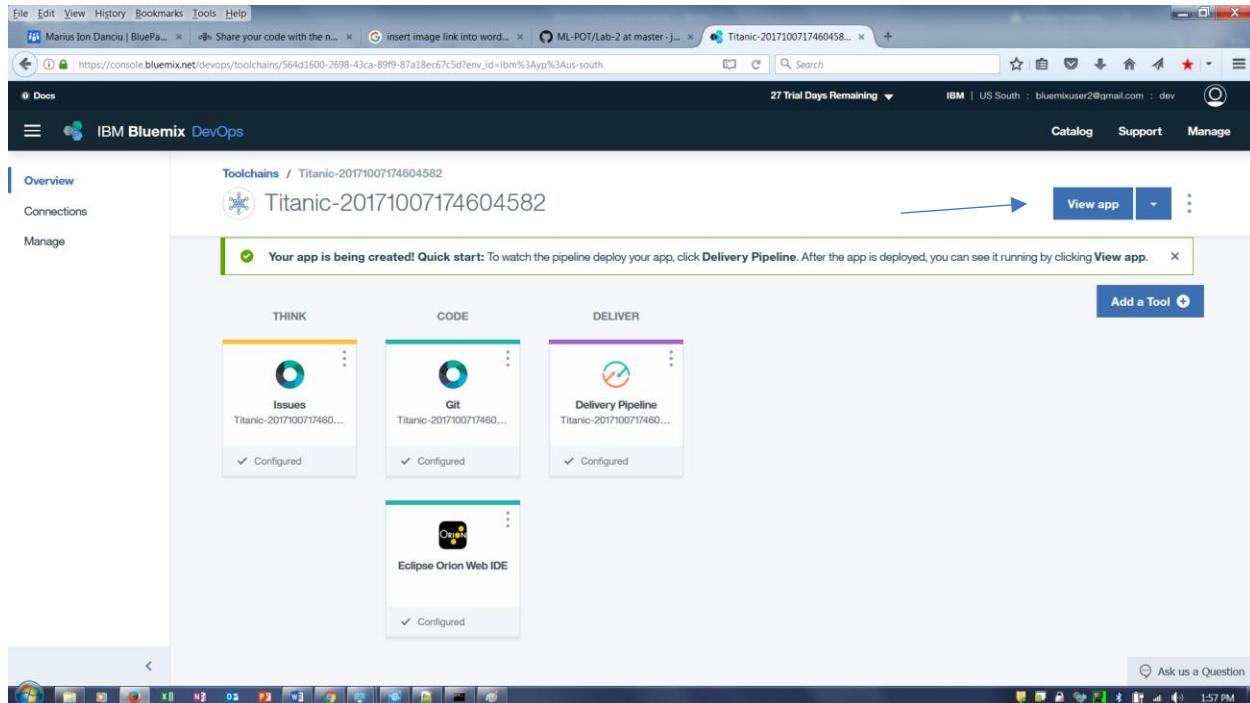
The screenshot shows the IBM Bluemix DevOps interface. At the top, there's a navigation bar with links for 'Docs', 'IBM Bluemix DevOps', 'Catalog', 'Support', and 'Manage'. Below the navigation, the main title is 'Deploy to Bluemix: Titanic app'. There's a note about deploying after clicking 'Deploy'. Under 'Tool Integrations', there are icons for 'Git Repos and Issue Tracking', 'Eclipse Orion Web IDE', and 'Delivery Pipeline'. A callout box over the 'Delivery Pipeline' icon says, 'The Delivery Pipeline automates continuous deployment.' It also shows an 'App name:' field containing 'Titanic-20171007174604582'. At the bottom right, there's a large blue 'Deploy' button. A blue arrow points from the text 'click **Delivery Pipeline**' in the instructions below to this 'Deploy' button.

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

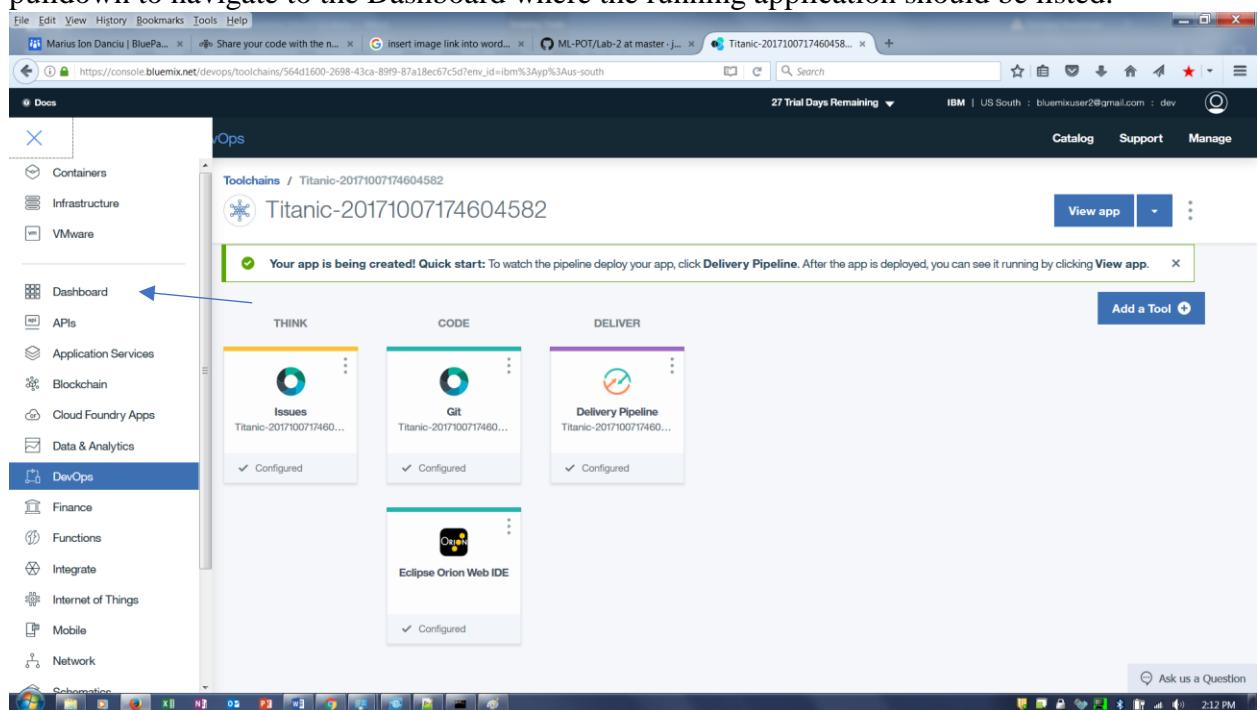


4. After the app is deployed successfully (should say Deployed passed now in the Deploy stage), return to the Delivery Pipeline by clicking on the vertical ellipse and click on View Toolchain.

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.



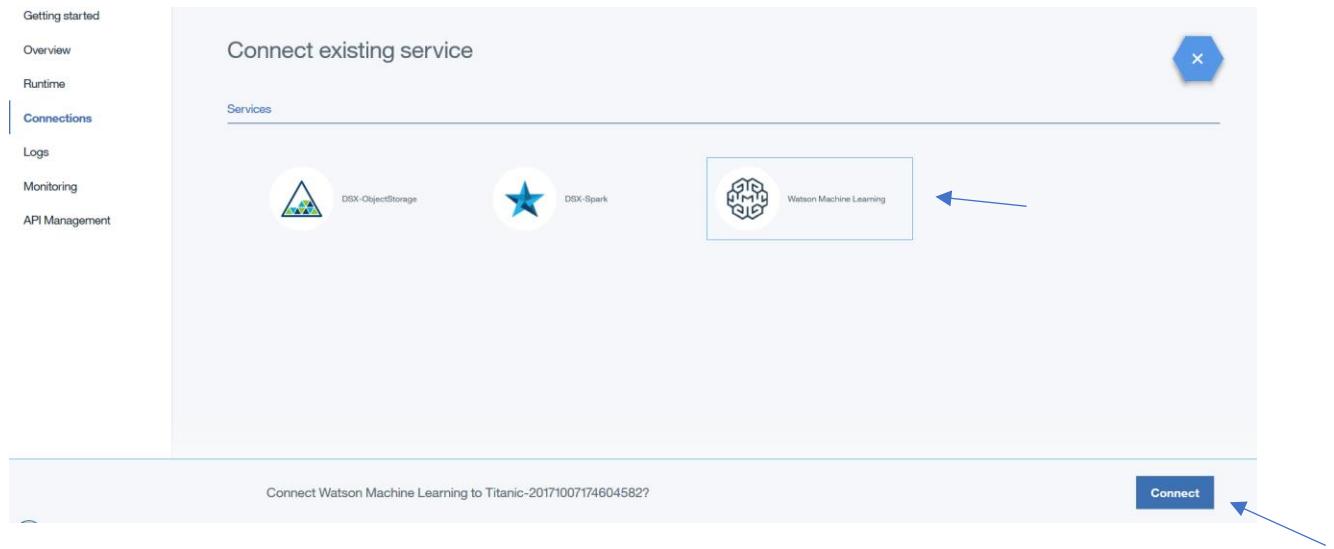
7. We are now going to connect the application to the Watson Machine Learning service that was created earlier. Click on the application name.

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.

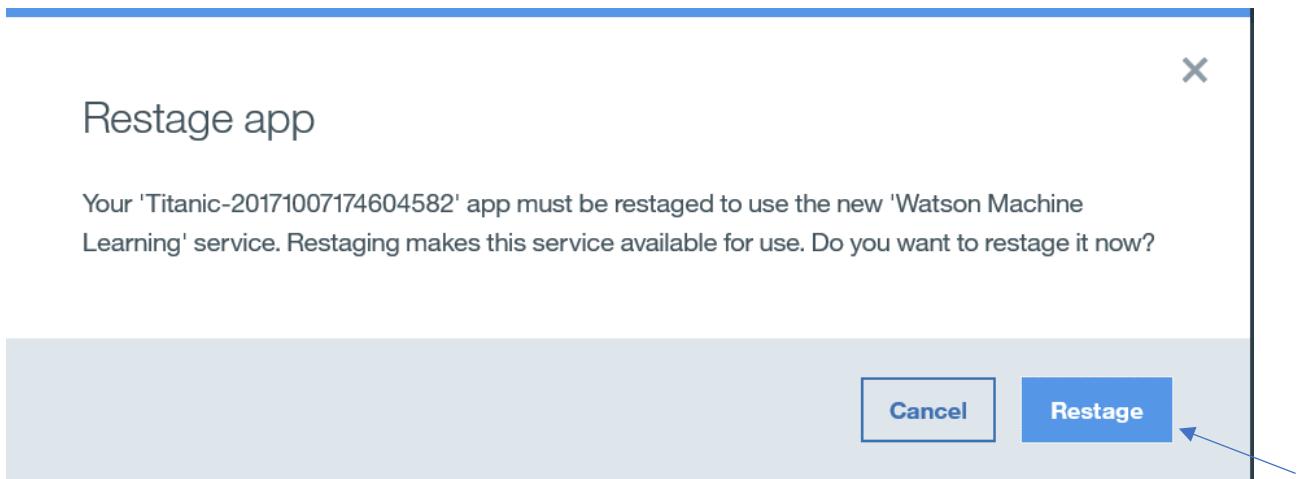
8. Scroll down until you see the Connections panel. Click on Connect Existing.

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.

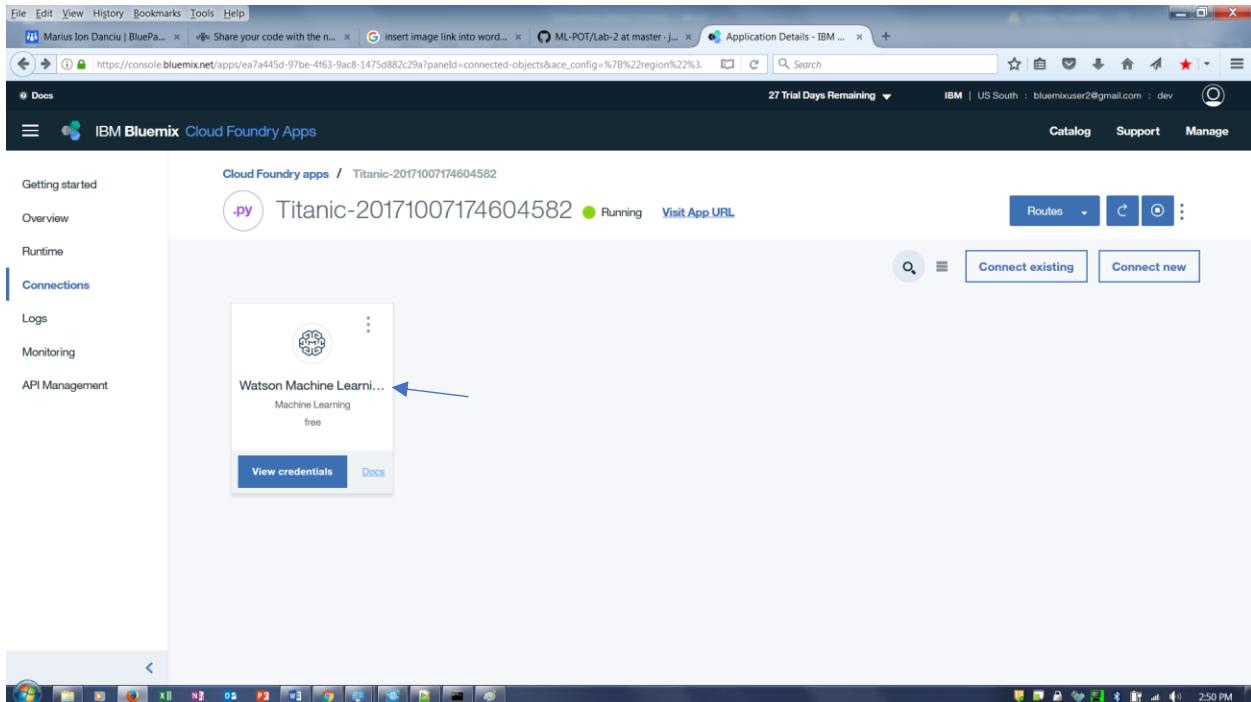
9. You should see at least 3 services listed. Two services are automatically provisioned for the Data Science Experience (Spark service, and Object Storage service). Click on the **Watson Machine Learning** service, and then click on **Connect**.



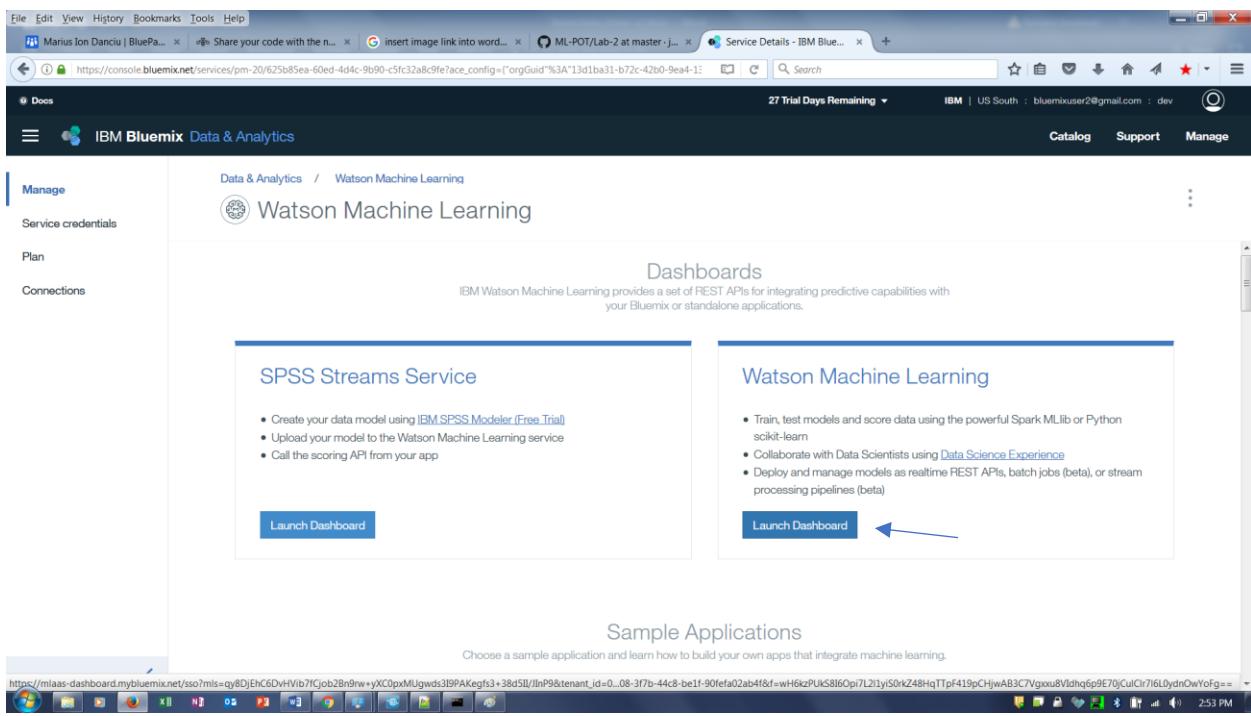
10. You will get a pop up that asks to Restage the application. Click on **Restage**.



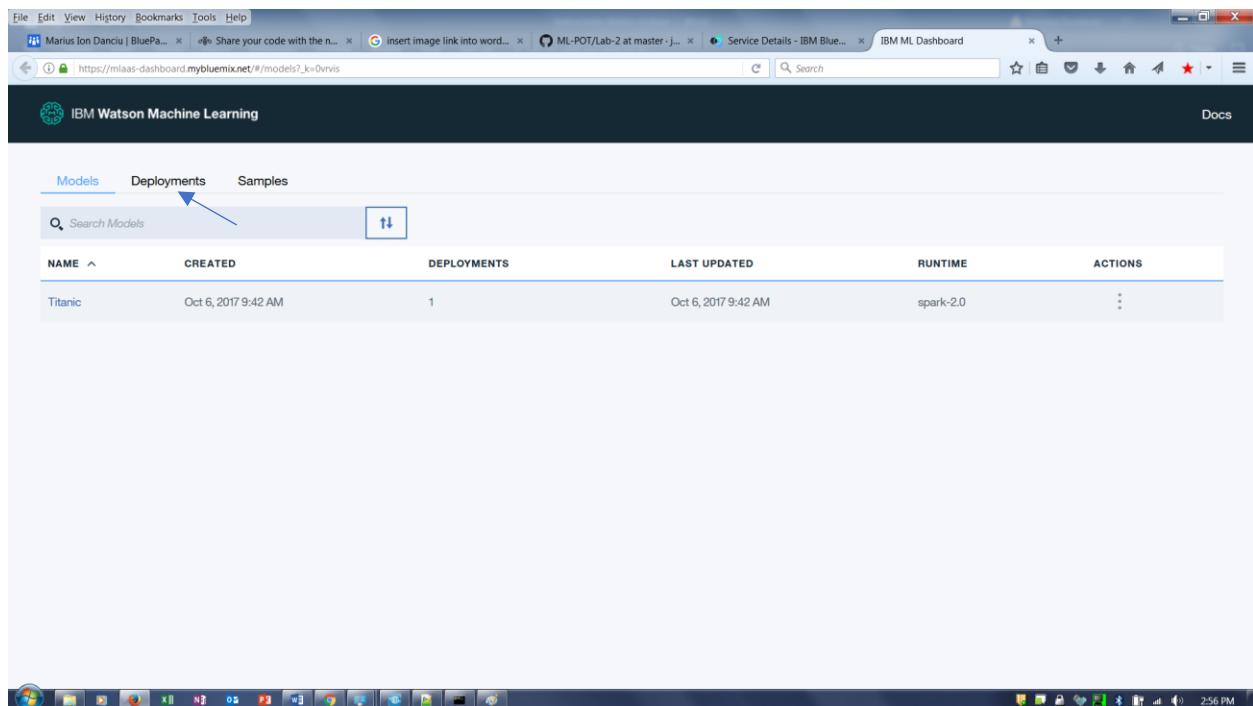
11. We now need to copy the scoring endpoint to the clipboard. Click on the **Watson Machine Learning** service.



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

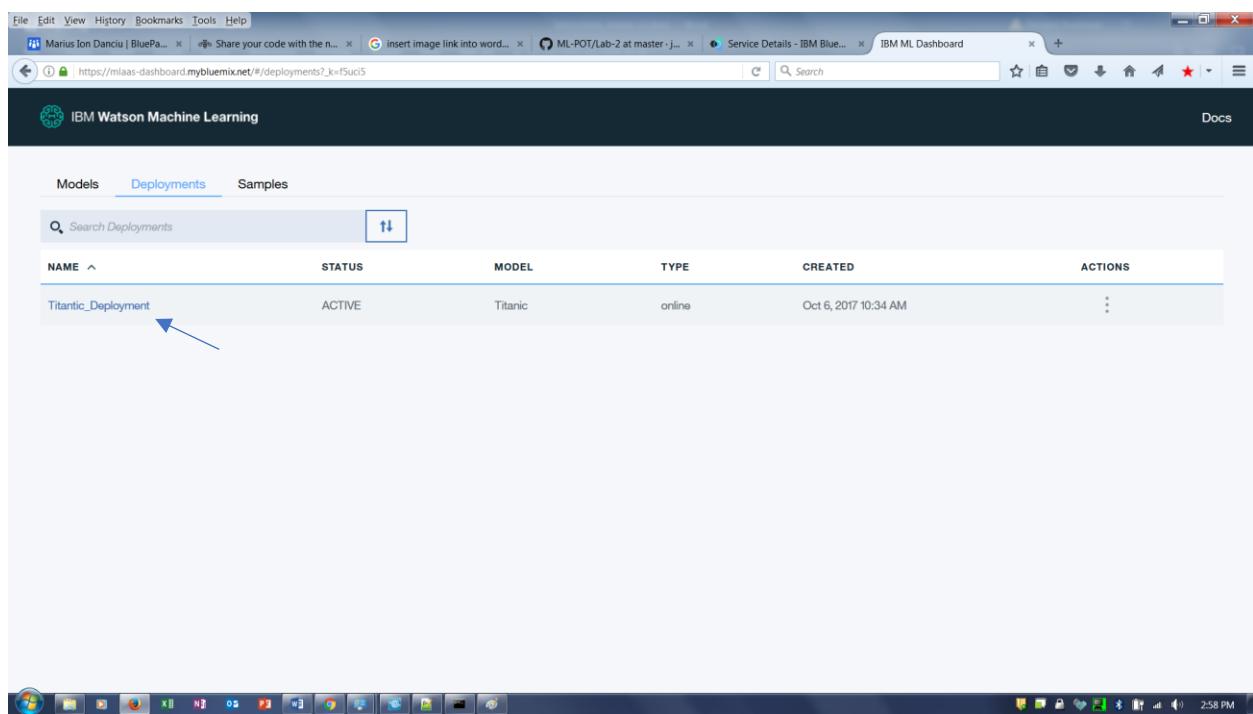


13. Click on the **Deployment** tab.



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 a three-dot 'Actions' menu. A blue arrow points to the 'Titanic_Deployment' row. The bottom of the screen shows a Windows taskbar.

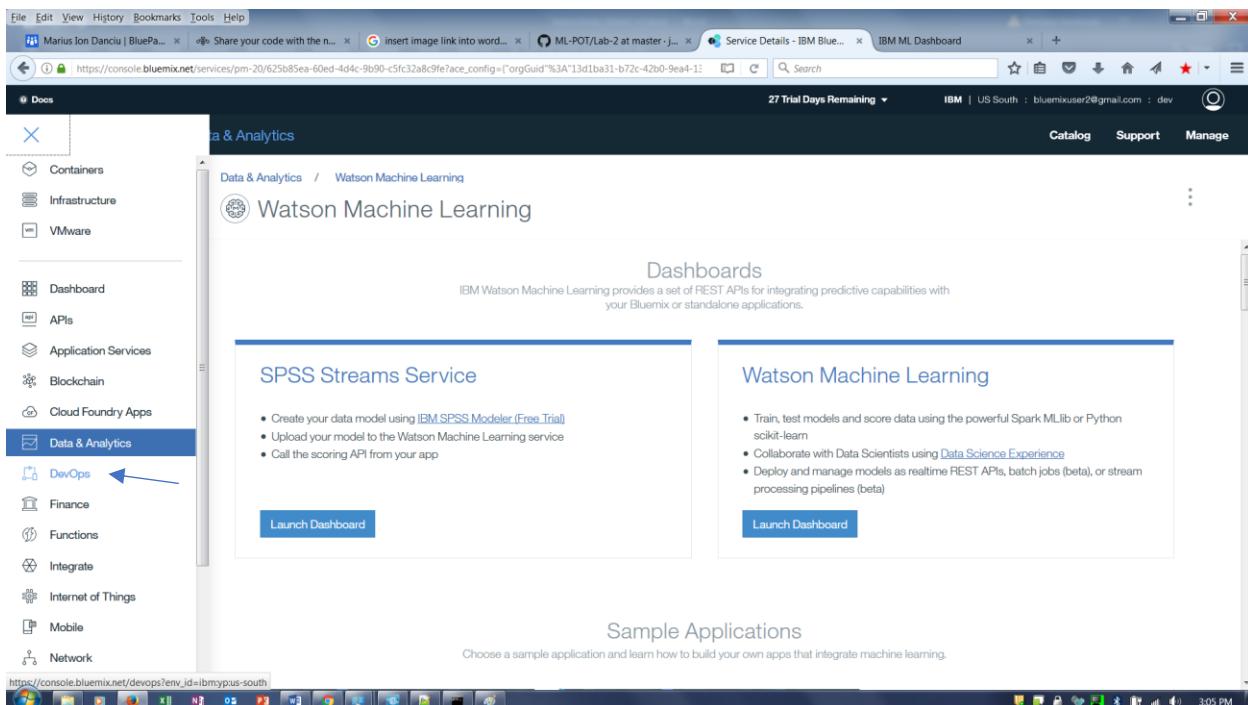
15. Scroll down to API details and copy the Scoring Endpoint value “http.....etc”

Deployment Name	Titanic_Deployment
Deployment Type	online
Deployment Id	7e591f82-e004-46e7-b35d-c9fb062ac8a
Deployment Status	ACTIVE
URL	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. In the top navigation bar, there are tabs for 'Service Details - IBM Blue...', 'ML-POT/Lab-2 at master', and 'IBM ML Dashboard'. Below the tabs, the status is shown as '27 Trial Days Remaining'. 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' category expanded, with 'DevOps' highlighted and a blue arrow pointing to it. Other options in the sidebar include 'Containers', 'Infrastructure', 'VMware', 'Dashboard', 'APIs', 'Application Services', 'Blockchain', 'Cloud Foundry Apps', 'Finance', 'Functions', 'Integrate', 'Internet of Things', 'Mobile', and 'Network'.

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

The screenshot shows the IBM Bluemix DevOps interface. The left sidebar has 'Toolchains' selected. The main area is titled 'Toolchains' and shows a table with one row: 'Titanic-20171007174604582'. A blue arrow points to this row. The top right has a 'Create a Toolchain' button.

18. Click on the Eclipse Orion Web IDE.

The screenshot shows the 'Toolchains / Titanic-20171007174604582' page. It displays three columns: 'THINK' (Issues), 'CODE' (Git), and 'DELIVER' (Delivery Pipeline). Below these is a fourth column labeled 'Eclipse Orion Web IDE'. A blue arrow points to the 'Eclipse Orion Web IDE' icon. A green box at the top says: 'Your app is being created! Quick start: To watch the pipeline deploy your app, click Delivery Pipeline. After the app is deployed, you can see it running by clicking View app.'

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.

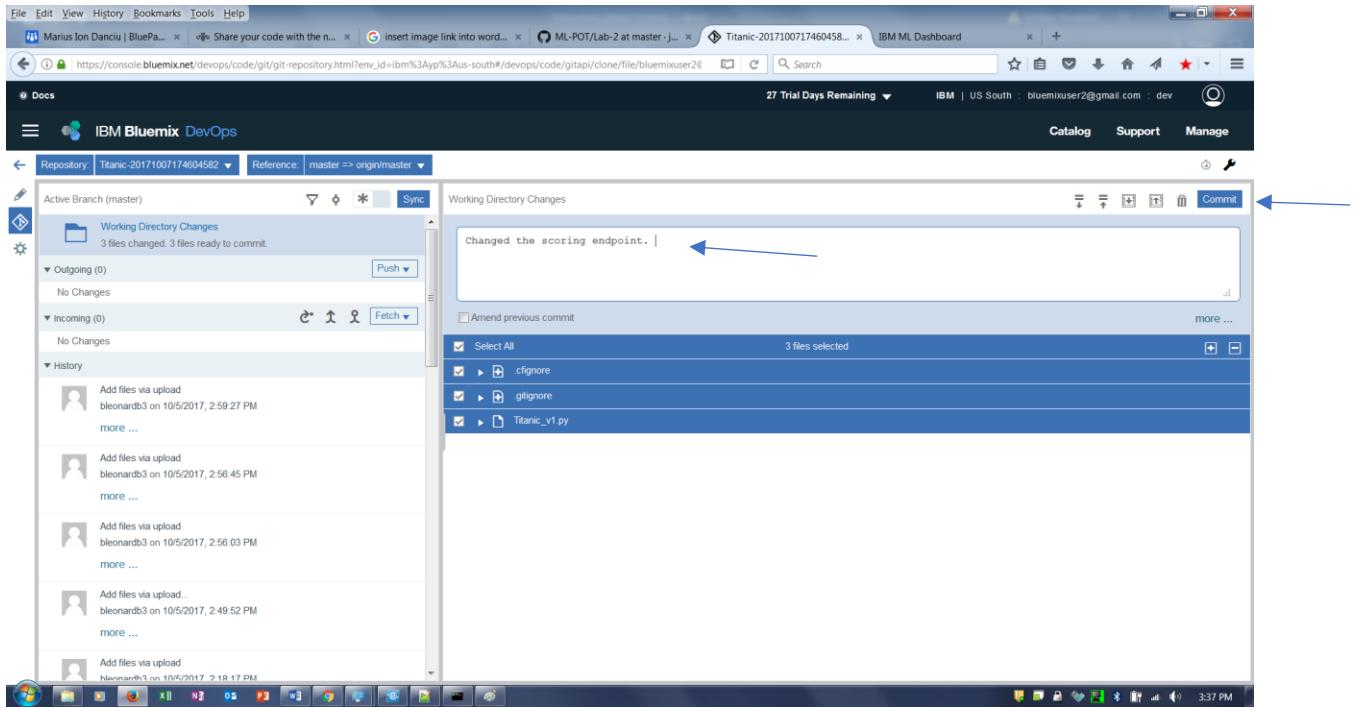
- static
- templates
- .cignore
- .gitignore
- LICENSE
- manifest.yml
- NOTICE
- Profile
- README.md
- requirements.txt
- Titanic_v1.py**

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.

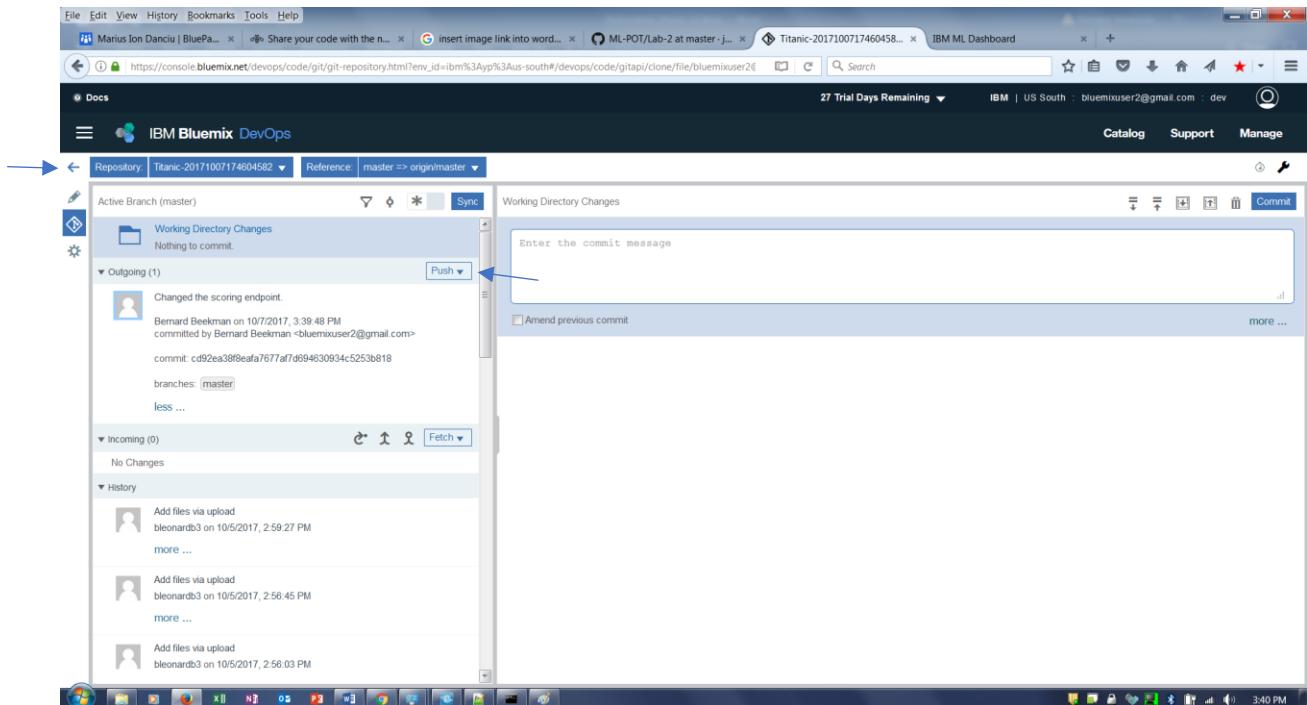
```

1 import urllib3, requests, json, os
2 app = Flask(__name__, static_folder='static')
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 = 'ec34cd2-3809-4378-a2f-e7e5981dd3d'
11
12 if 'VCAP_SERVICES' in os.environ:
13     vcap = json.loads(os.getenv('VCAP_SERVICES'))
14     print('Found VCAP_SERVICES')
15     if 'wml' in vcap:
16         creds = vcap['wml'][0]['credentials']
17         username = creds['username']
18         password = creds['password']
19         url = creds['url']
20
21 scoring_endpoint = url + 'v3/wml_instances/625b85ea-60ed-4d4c-9b90-c5fc32a8c9fe/published_models/0788b520-0f19-4f7c-9ee5-03a9f7ed899d'
22 app.config['SECRET_KEY'] = 'secretpassw0rd'
23 bootstrap = Bootstrap(app)
24 class TitanicForm(FlaskForm):
25     pclass = RadioField('Passenger Class:', coerce=int, choices=[('1','First'),('2','Second'),('3','Third')])
26     name = StringField('Name:')
27     sex = RadioField('Sex:', coerce=str, choices=[('male','Male'),('female','Female')])
28     age = RadioField('Age:', coerce=int, choices=[('0','0-5'),('1','6-11'),('2','12-17'),('3','18-39'),('4','40-64'),('5','65-79'),('6','80+')])
29     ticket = StringField('Ticket:')
30     fare = FloatField('Fare:')
31     sibsp = IntegerField('Number of siblings/spouses:')
32     parch = IntegerField('Number of parents/children:')
33     embarked = RadioField('Embarked location:', coerce=str, choices=[('S','South Hampton'),('C','Cherbourg'),('Q','Queenstown')])
34     submit = SubmitField('Submit')
35     @app.route('/', methods=['GET', 'POST'])
36     def index():
37         form = TitanicForm()
38         if form.is_submitted():
39             sex = form.sex.data
40             name = form.name.data
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

Embarc 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!](#)