

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

My Projects > LMCO

Overview Analytics Assets Data Assets Bookmarks Deployments Collaborators Settings

Project Information

Project Name: LMCO

Description: Project Description

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 **New**. Select the **Lite (Free)** instance and **Create**

IBM Watson Data Platform

Machine Learning

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. The Machine Learning service is a set of REST APIs that you can call from any programming language 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.

Spark and Python Machine Learning features
Take advantage of Spark MLlib and scikit-learn machine learning models management and deployment - online, batch and streaming.

Integration with Data Science Experience
Visit <http://datascience.ibm.com>. Create and train predictive analytics models with the best tools and the latest expertise in a social environment built by data scientists.

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

Plan Features Pricing

Lite 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 Creation

Organization: jpatter@us.ibm.com

Plan
Lite

Space
dev

Service Name
MyMachineLearning

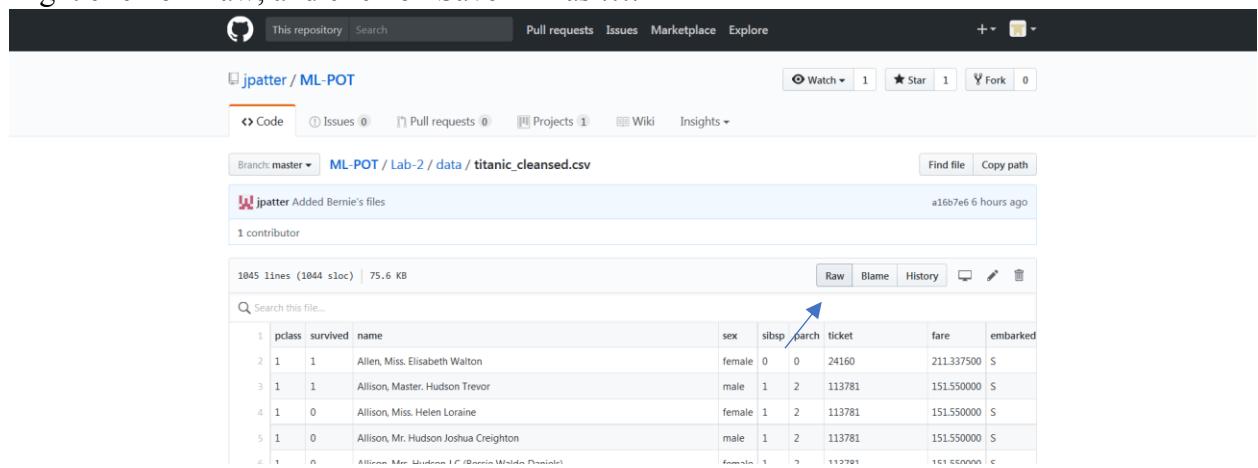


Step 2: Adding a Data Asset to the project

1. Download the Titanic data file from [Cleansed Titanic Data Set](#)

The data in this file has already been prepared and it is 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 'Code' tab is selected, showing the file 'ML-POT / Lab-2 / data / titanic_cleaned.csv'. The file was last updated by 'jpatter' 6 hours ago. The file contains 1045 lines (1044 sloc) and is 75.6 KB. A preview of the CSV data shows columns: pclass, survived, name, sex, sibsp, parch, ticket, fare, embarked. The first few rows show passengers with their details and survival status. An arrow points to the 'Raw' link in the file preview header.

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	c

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

The screenshot shows the IBM Data Science Experience interface. At the top, there are several tabs and a search bar. Below the header, the main navigation bar includes 'Projects', 'Catalog BETA', 'Tools', 'Data Services', and 'Community'. Under 'My Projects > DSX Labs', there are two sections: 'Notebooks' and 'Data Assets'. The 'Notebooks' section lists two entries: 'Machine Learning with Spark - Joel' and 'Connect and Interact with DashDB'. The 'Data Assets' section lists two entries: 'Female-human-trafficking.csv' and 'myDashDB'. At the bottom right of each section is a blue button labeled '+ add [asset type]'. A blue arrow points from the text 'click the "add [asset type]" button' to the '+ add notebooks' button in the 'Notebooks' section. Another blue arrow points to the '+ add data assets' button in the 'Data Assets' section.

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

The screenshot shows the IBM Watson Data Platform interface. At the top, there are tabs for 'Projects', 'Tools', 'Catalog BETA', 'Data Services', and 'Community'. Below the header, the main navigation bar includes 'My Projects > Test'. Under the 'Assets' tab, there is a search bar and a section titled 'Data assets'. A large callout box highlights the 'Load' button, which has a dashed box around it and a blue arrow pointing to it. The text 'Drop file here or browse your files to add a new file' is displayed below the button.

Step 3: Create a Model to predict survival

- Click on the **Assets** Tab

IBM Watson Data Platform

My Projects > Test

Overview Assets Bookmarks Deployments Collaborators Settings

What assets are you looking for?

Data assets

NAME	TYPE	SERVICE	LAST MODIFIED	ACTIONS
titanic_cleansed.csv	CSV	Object Storage (Swift API)		⋮

Load Files Connections

Find in storage

0 selected

titanic_cleansed.csv

2. Click on add models.

IBM Watson Data Platform

My Projects > Test

Overview Assets Bookmarks Deployments Collaborators Settings

What assets are you looking for?

Data assets

NAME	TYPE	SERVICE	LAST MODIFIED	ACTIONS
titanic_cleansed.csv	CSV	Object Storage (Swift API)		⋮

Notebooks

NAME	SHARED	SCHEDULED	STATUS	LANGUAGE	LAST EDITOR	LAST MODIFIED	ACTIONS
Lab1Student				Python 2.7	Joel Patterson	30 Nov 2017	⋮
Lab3Student				Python 3.5	Joel Patterson	19 Sep 2017	⋮
Lab2Student				Python 2.7	Joel Patterson	18 Sep 2017	⋮

Models

NAME	STATUS	RUNTIME	LAST MODIFIED	ACTIONS
you currently have no models				

Load Files Connections

Find in storage

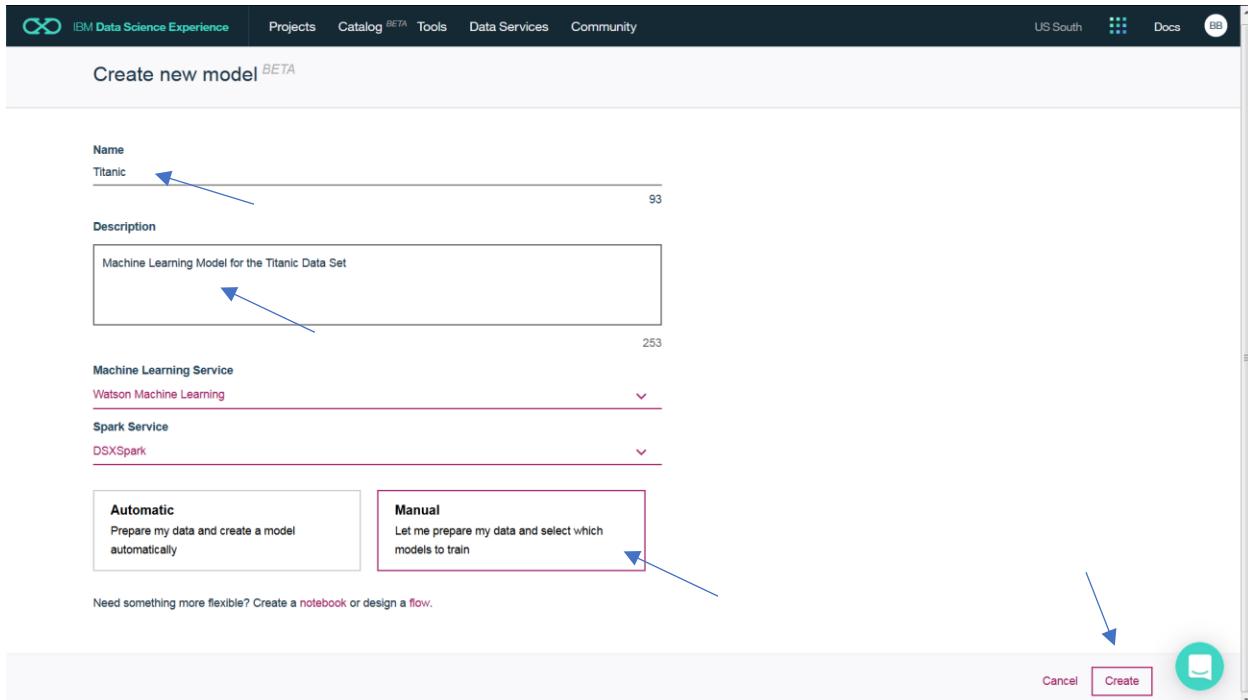
0 selected

titanic_cleansed.csv

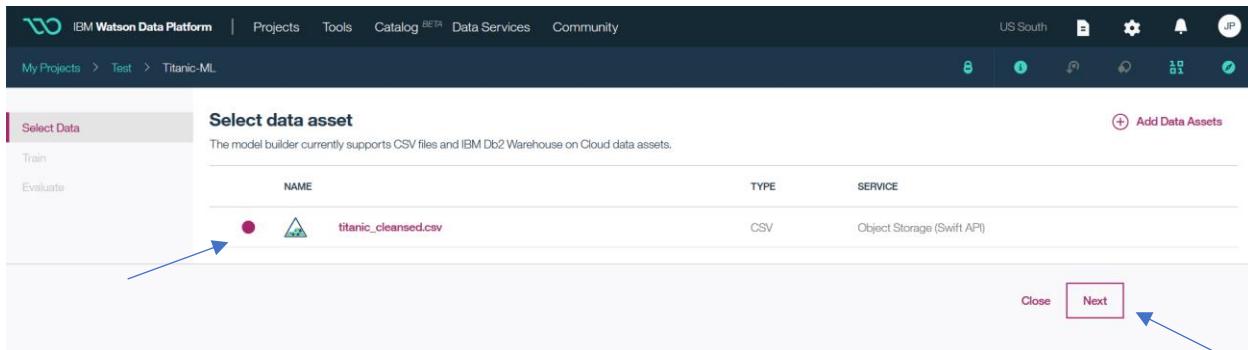
New notebook

New model

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



4. Click on the titanic_cleaned.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.

File Edit View History Bookmarks Tools Help

Mail Service Details - yourIBM Jewish Holidays 2... Proof-of-Tech... ML-POT/titanic.cs... IBM Data Science ... The Largest 4G LT...

https://apisportal.ibm.com/ml/model-builders/67159323-40a4-48d8-911c-a6934abfbda0/builder/configure?projectId=8e5d9ac9-05cc-4b93-a857-ad...

Search

IBM Data Science Experience Projects Catalog BETA Tools Data Services Community US South Docs BB

My Projects > DSX Labs > Titanic

Select Data Train Evaluate 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

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

File Edit View History Bookmarks Tools Help

Mail Service Details - yourIBM Jewish Holidays 2... Proof-of-Tech... ML-POT/titanic.cs... IBM Data Science ... The Largest 4G LT...

https://apisportal.ibm.com/ml/model-builders/67159323-40a4-48d8-911c-a6934abfbda0/builder/configure?projectId=8e5d9ac9-05cc-4b93-a857-ad...

Search

IBM Data Science Experience Projects Catalog BETA Tools Data Services Community US South Docs BB

My Projects > DSX Labs > Titanic

Select Data Train Evaluate Select estimator(s)

What type of estimator are you looking for?

Logistic Regression
Analyzes a data set in which there are one or more independent variables that determine one of two outcomes. Only binary...

Decision Tree Classifier
Maps observations about an item (represented in the branches) to conclusions about the item's target value (represented in...

Random Forest Classifier
Constructs multiple decision trees to produce the label that is a mode of each decision tree. It supports both binary and...

Gradient Boosted Tree Classifier
Produces a classification prediction model in the form of an ensemble of decision trees.

Train: 60 Test: 20 Holdout: 20

Cancel Add

7. Select the **Next** button.

Select a technique

You cannot change label column, feature columns, model type, or validation split after adding an estimator. You must first delete all estimators in order to make changes to these attributes.

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

Feature columns
pclass (Integer), sex (String), sibsp (Integer), parch (Integer), fare (Decimal), emb

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

Configured estimators

- Logistic Regression (Not Yet Trained)
- Decision Tree Classifier (Not Yet Trained)
- Random Forest Classifier (Not Yet Trained)
- Gradient Boosted Tree Classifier (Not Yet Trained)

Close Previous Next

- 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** (if it is the best) and then click **Save**.

ESTIMATOR TYPE	STATUS	PERFORMANCE	AREA UNDER ROC CURVE	AREA UNDER PR CURVE	LAST EVALUATION	ACTIONS
LogisticRegression	Trained & Evaluated	Good	0.88036	0.87011	30 Nov 2017, 4:31 PM	⋮
RandomForestClassifier	Trained & Evaluated	Good	0.87617	0.8728	30 Nov 2017, 4:32 PM	⋮
GBTClassifier	Trained & Evaluated	Good	0.8003	0.81388	30 Nov 2017, 4:33 PM	⋮
DecisionTreeClassifier	Trained & Evaluated	Fail	0.4506	0.49969	30 Nov 2017, 4:32 PM	⋮

Close Previous Save

- The system displays the model training summary.

The screenshot shows the IBM Watson Data Platform interface. At the top, there's a navigation bar with links for Projects, Tools, Catalog (BETA), Data Services, and Community. On the right side of the header, there are icons for US South, a file, settings, a bell, and a user profile. Below the header, the path 'My Projects > Test > Titanic' is visible. The main content area is titled 'Titanic' with a trash icon. It has three tabs: Overview (selected), Evaluation, and Deployments. Under the 'Overview' tab, there's a 'Summary' section with the following data:

Machine learning service	ML-113017
Runtime environment	spark-2.0
Training date	30 Nov 2017, 4:39 PM
Label column	survived
Latest version	e00e9080-f42c-4e68-b45f-3430c7b86e90
Model builder details	View

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 'Deployments' tab selected in the 'Titanic' project. A blue arrow points from the text 'Select the Deployments Tab' to the 'Deployments' tab in the navigation bar. Another blue arrow points from the text 'Click on Add Deployments' to the '+ Add Deployment' button in the top right corner of the deployment table. The table has columns: NAME, STATUS, DEPLOYMENT TYPE, and ACTIONS. A message at the bottom says 'Your model is not deployed.'

3. Select **Web Service** tab, enter **Titanic_Deployment** for **Name**, and click on **Save**.

Create Deployment

Web Service Batch Prediction Real-time Streaming Predictions

Name
Titanic_Deployment

Description
Web Service Deployment Description

300

Cancel Save

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

My Projects > Test > Titanic

Titanic

NAME	STATUS	DEPLOYMENT TYPE	ACTIONS
Titanic_Deployment	ACTIVE	Web Service	

Add Deployment

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** to test out the API.

My Projects > Test > Titanic > Titanic_Deployment

Titanic_Deployment

Scoring End-point	View API Specification
https://ibm-watson-ml.mybluemix.net/v3/wml_instances/c558a320-34aa-4e19-be3b-ca11f27621db/published_models/4fe76ba5-0f51-4551-af48-6e1d5f2db83e/deployments/43992e42-80bb-4139-be76-4bcff653b6bb6/online	See code snippets below for information on how to retrieve the WML Authorization Token to be passed with scoring requests.
Authorization: Bearer <token>	Content-type: application/json
Required if the request body is sent in JSON format.	

Code Snippets

- 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.

Titanic_Deployment

Overview Implementation Test

Input data

pclass
1

name
Allen, Miss. Elisabeth Walton

sex
female

sibsp
0

Predict

- The predicted result is returned.

Titanic_Deployment

Overview Implementation Test

Input data

pclass
1

name
Allen, Miss. Elisabeth Walton

sex
female

sibsp
0

Predict

Predicted value for survived

Value	Percentage
1	92.96%
0	7.04%

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.

The screenshot shows the 'Deploy to Bluemix' interface. At the top, there is a note about deployment and a toolchain name input field ('Toolchain Name: Titanic-20171130220547025'). Below that, 'Select Region:' is set to 'US South' and 'Choose an organization:' is set to 'ipatter@us.ibm.com'. Under 'Tool Integrations', there are three icons: 'Git Repos and Issue Tracking', 'Eclipse Orion Web IDE', and 'Delivery Pipeline' (which is highlighted with a blue border). A tooltip for the Delivery Pipeline says: 'The Delivery Pipeline automates continuous deployment.' At the bottom, there is an 'App name:' input field ('Titanic-20171130220547025'), a 'Region' dropdown ('US South (Production)'), an 'Organization' dropdown ('ipatter@us.ibm.com'), and a 'Space' dropdown ('dev'). A large blue 'Deploy' button is at the bottom right, with a blue arrow pointing towards it from the right side of the image.

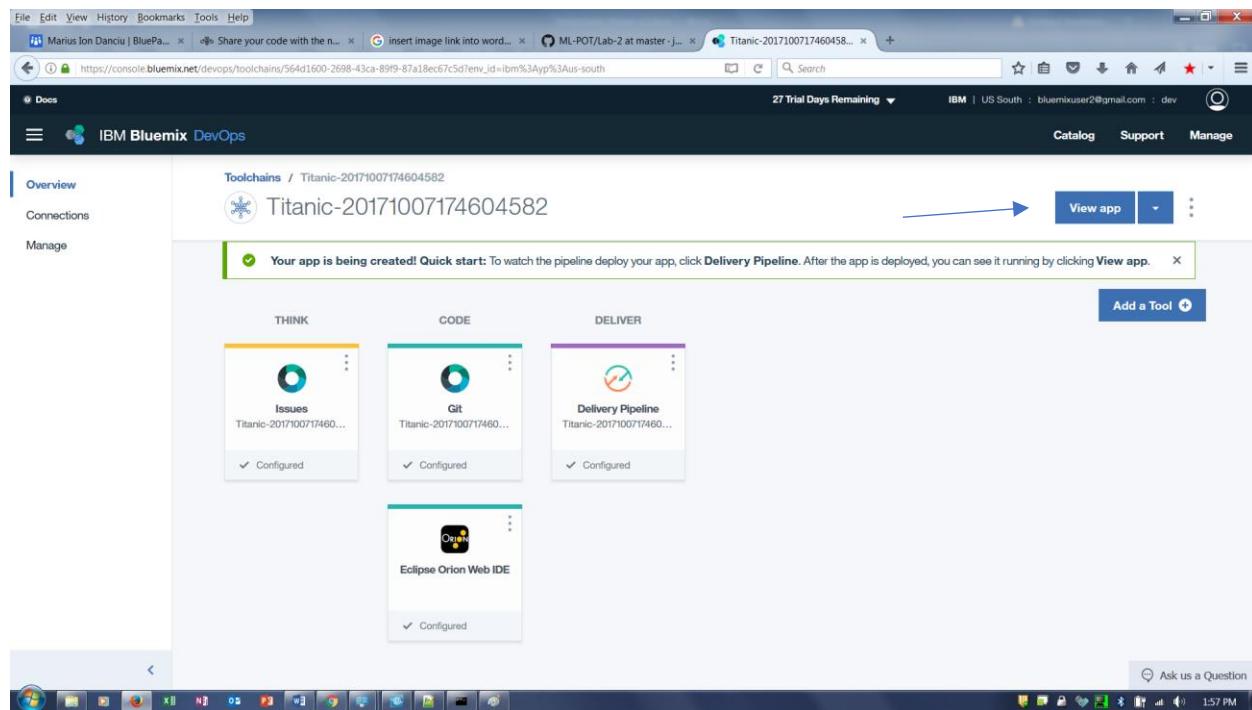
3. Your app is being created! To watch the pipeline deploy your app, click **Delivery Pipeline**.

The screenshot shows the IBM Bluemix DevOps interface. The left sidebar has 'Overview', 'Connections', and 'Manage' options. The main area is titled 'Toolchains / Titanic-20171007174604582'. It displays three stages: 'THINK', 'CODE', and 'DELIVER'. The 'DELIVER' stage is highlighted with a blue arrow pointing to it. The stage card shows 'Delivery Pipeline' is configured. Below the stages is a section for 'Eclipse Orion Web IDE' which is also configured. At the bottom right of the main area is a 'View app' button.

4. After the app is deployed successfully (should say Deploy Passed 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 application 'Titanic-20171007174604582'. The page is divided into 'Build Stage' and 'Deploy Stage'. The 'Deploy Stage' shows a green 'STAGE PASSED' status. The 'Jobs' section indicates 'Deploy Passed now'. The 'LAST EXECUTION RESULT' section shows a successful build named 'Build 1'. At the top right of the page, there is a vertical ellipsis icon 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.

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**.

THINK	CODE	DELIVER
Issues Titanic-20171007174604582... ✓ Configured	Git Titanic-20171007174604582... ✓ Configured	Delivery Pipeline Titanic-20171007174604582... ✓ Configured
Eclipse Orion Web IDE ✓ Configured		

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

Cloud Foundry Apps (1) 128 MB/2 GB Used						
NAME	ROUTE	MEMORY (MB)	INSTANCES	RUNNING	STATE	ACTIONS
Titanic-20171007174604582	titanic-20171007174604582-jackleg-pirogue.myblu...	128	1	1	Running	

Services (5) 5/10 Used						
NAME	SERVICE OFFERING	PLAN	ACTIONS			
Continuous Delivery	Continuous Delivery	Free (Deprecated)				
Data Science Experience-gz	Data Science Experience	Lite				
DSX-ObjectStorage	Object Storage	Lite				
DSX-Spark	Apache Spark	Lite				
Watson Machine Learning	Machine Learning	Free				

- Scroll down until you see the Connections panel. Click on **Create Connection**.

The screenshot shows the IBM Cloud Overview page for a Cloud Foundry app named "Titanic-20171130220547025". The app is running in the US South location, Space dev, and is built with Python. It has 1 instance with 128 MB memory per instance, totaling 128 MB allocation. The runtime cost is \$0.00 for the current period and estimated at \$0.00 for the entire billing period (Nov 1, 2017 - Nov 30, 2017). A blue arrow points from the "Create connection" button in the Connections section to the "View full usage details" button in the Runtime cost section.

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 for your application, and then click on **Connect**.

The screenshot shows the IBM Cloud Connections page titled "Connect Existing Compatible Service". It lists 14 items, with a blue arrow pointing to the "Machine Learning" service entry. The table columns are SERVICES, RESOURCE GROUP, PLAN, and SERVICE OFFERING.

SERVICES	RESOURCE GROUP	PLAN	SERVICE OFFERING
Apache Spark-Weather	--	Lite	Apache Spark
Apache Spark-Weather_objectstore	--	Lite	Object Storage
cloud-object-storage-joel	default	Lite	Cloud Object Storage
Decision Optimization-hz	--	Beta - 10 core / 60 GB (ODSTRIAL)	Decision Optimization
DSE-Spark	--	Lite	Apache Spark
DSE-Spark20	--	Lite	Apache Spark
JoelDashDB	--	Entry	Db2 Warehouse on Cloud
ML	--	Lite	Machine Learning
ML-113017	--	Lite	Machine Learning
ML-POT	--	Lite	Apache Spark

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

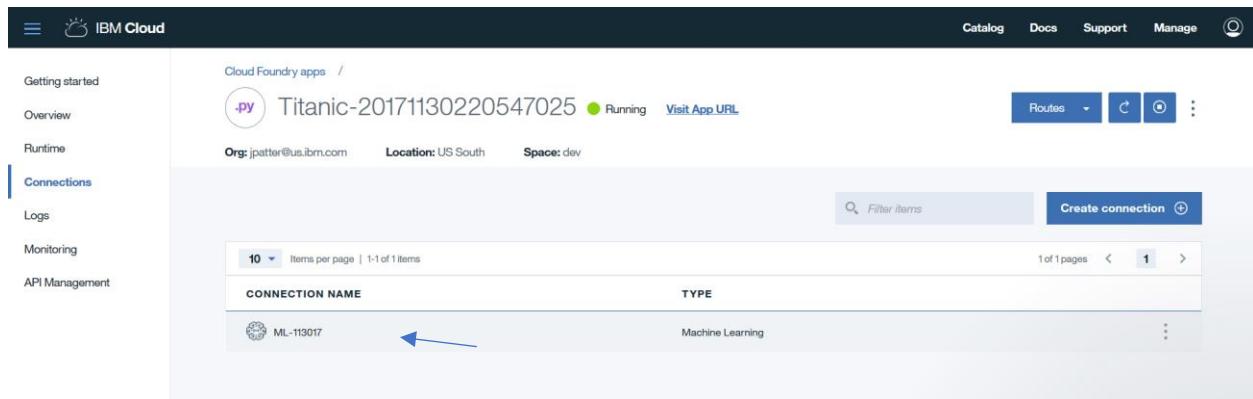
Restage app

Your 'Titanic-20171007174604582' app must be restaged to use the new 'Watson Machine Learning' service. Restaging makes this service available for use. Do you want to restage it now?

Cancel **Restage**

←

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



The screenshot shows the IBM Cloud dashboard with the 'Connections' section selected. On the right, the 'Watson Machine Learning' service is listed under 'Cloud Foundry apps'. The service name is 'Titanic-20171130220547025', status is 'Running', and it's located in 'US South' space 'dev'. Below the service details, there's a table showing a single connection:

CONNECTION NAME	TYPE
ML-113017	Machine Learning

A blue arrow points to the 'ML-113017' connection name.

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

The screenshot shows the IBM Bluemix Data & Analytics interface. On the left sidebar, under the 'Manage' section, there are links for 'Service credentials', 'Plan', and 'Connections'. The main content area is titled 'Watson Machine Learning'. It features two sections: 'Dashboards' and 'Watson Machine Learning'. The 'Dashboards' section contains a box for 'SPSS Streams Service' with three bullet points: 'Create your data model using IBM SPSS Modeler (Free Trial)', 'Upload your model to the Watson Machine Learning service', and 'Call the scoring API from your app'. Below this is a 'Launch Dashboard' button. The 'Watson Machine Learning' section has a list of bullet points: 'Train, test models and score data using the powerful Spark MLlib or Python scikit-learn', 'Collaborate with Data Scientists using Data Science Experience', and 'Deploy and manage models as realtime REST APIs, batch jobs (beta), or stream processing pipelines (beta)'. Below this is another 'Launch Dashboard' button. A blue arrow points from the 'Launch Dashboard' button in the 'Watson Machine Learning' section towards the right edge of the screen.

13. Click on the Deployment tab.

The screenshot shows the IBM Watson Machine Learning dashboard. At the top, there are tabs for 'Models', 'Deployments' (which is highlighted with a blue arrow), and 'Samples'. Below the tabs is a search bar labeled 'Search Models' and a refresh icon. The main content area displays a table of models. The columns are: NAME, CREATED, DEPLOYMENTS, LAST UPDATED, RUNTIME, and ACTIONS. There is one entry in the table:

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. The table lists one deployment:

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”

The screenshot shows the deployment details screen. Under 'API Details', the 'Scoring Endpoint' field contains the URL: https://ibm-watson-ml.mybluemix.net/v3/wml_instances/625b85ea-60ed-4d4c-9b90-c5fc32a8c9fe/published_models/0788b520-0f19-4f7c-9ee5-03a9f7ed899d/deployments/7e591f82-e004-46e7-b35d-c9fb062ac8a/online

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.

Watson Machine Learning

SPSS Streams Service

- Create your data model using [IBM SPSS Modeler \(Free Trial\)](#)
- Upload your model to the Watson Machine Learning service
- Call the scoring API from your app

Watson Machine Learning

- Train, test models and score data using the powerful Spark MLlib or Python scikit-learn
- Collaborate with Data Scientists using [Data Science Experience](#)
- Deploy and manage models as realtime REST APIs, batch jobs (beta), or stream processing pipelines (beta)

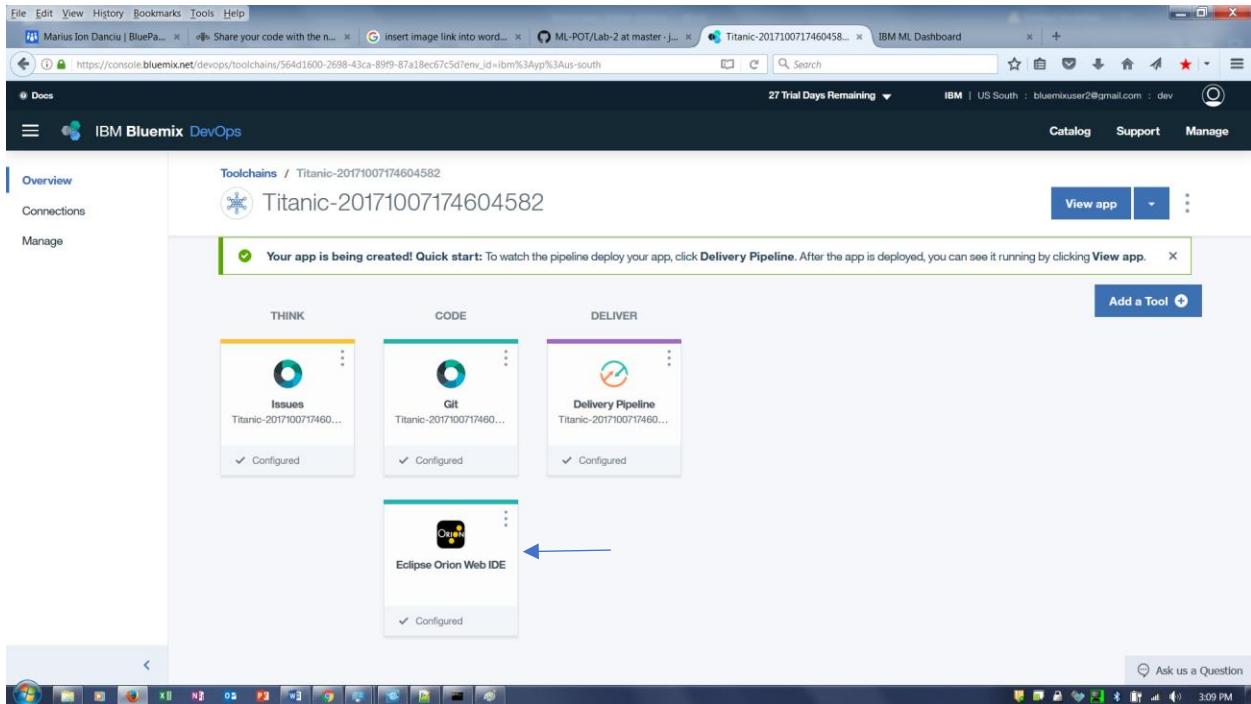
Sample Applications

Choose a sample application and learn how to build your own apps that integrate machine learning.

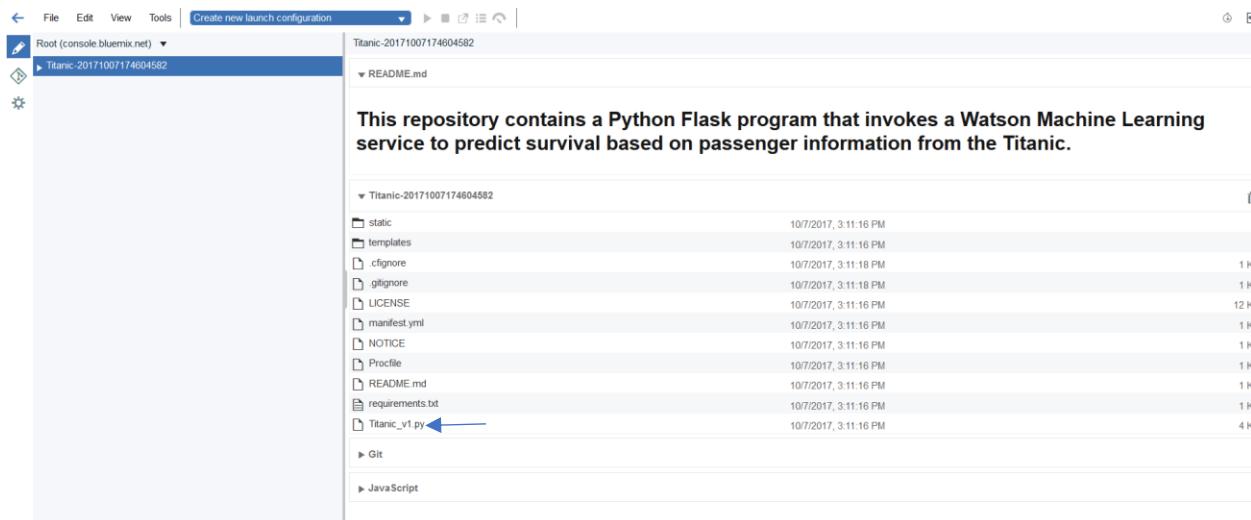
17. We are now going to paste the scoring endpoint into the application code. Click on the Toolchain.

NAME	TOOL INTEGRATIONS
Titanic-20171007174604582	

18. Click on the Eclipse Orion Web IDE.

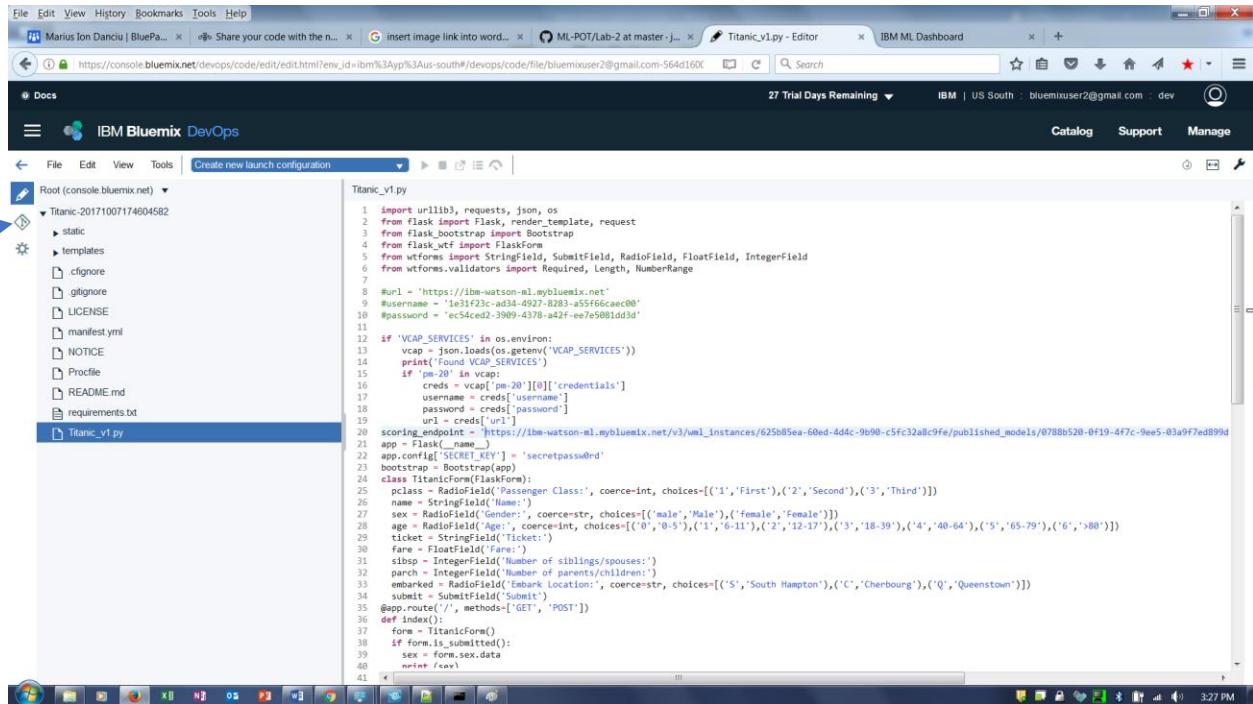


19. Click on the Titanic_v1 file.



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 or File >

Save to save the file. Then click on the  icon on the top left.

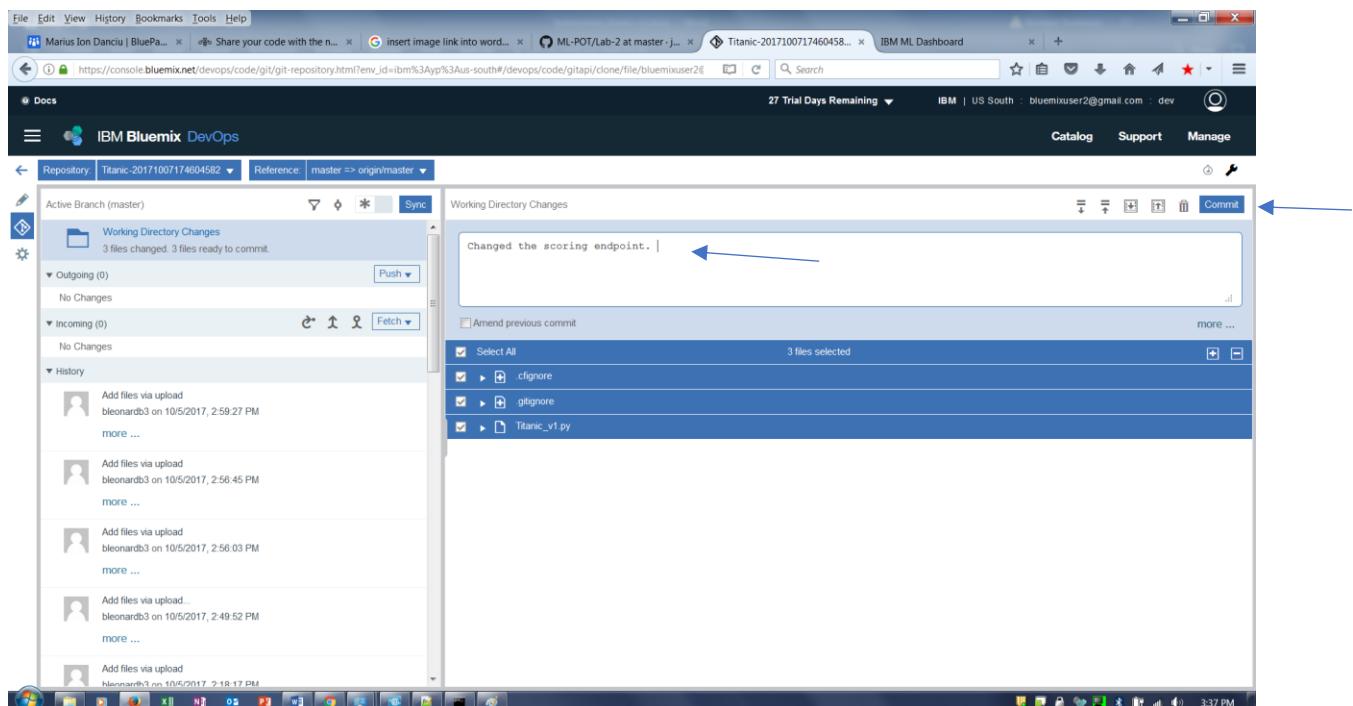


```

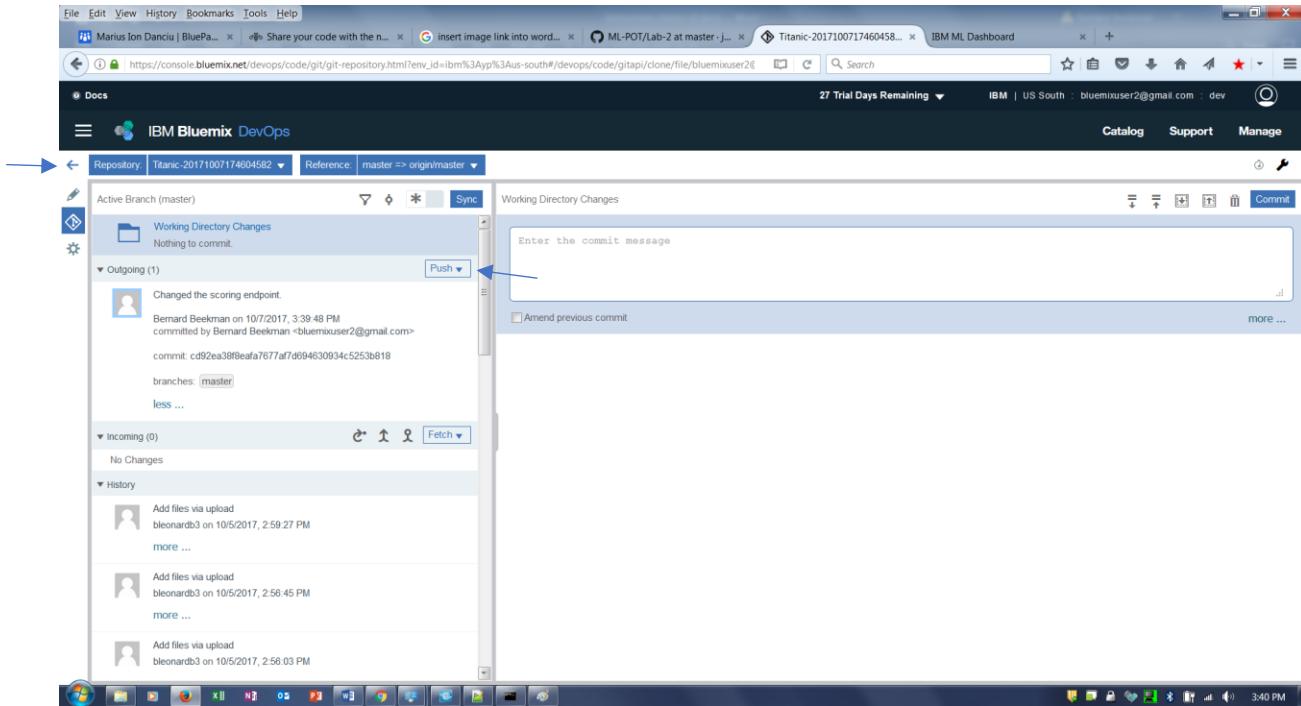
1 import urllib3, requests, json, os
2 from flask import Flask, render_template, request
3 from flask_bootstrap import Bootstrap
4 from flask_wtf import FlaskForm
5 from wtforms import StringField, SubmitField, RadioField, FloatField, IntegerField
6 from wtforms.validators import Required, Length, NumberRange
7
8 #url = "https://ibm-watson-ml.mybluemix.net"
9 #username = "1e1f1f23c-ad34-4927-8283-a55ff6caec00"
10 #password = "ec54cd2-3909-4378-a42f-ee7e5081dd3d"
11
12 if 'VCAP_SERVICES' in os.environ:
13     vcap = json.loads(os.getenv('VCAP_SERVICES'))
14     port = [found['credentials'] for found in vcap['app']]
15     if 'pm-20' in vcap:
16         creds = vcap['pm-20'][0]['credentials']
17         username = creds['username']
18         password = creds['password']
19         url = creds['url']
20     scoring_endpoint = "[REDACTED]"
21     app = Flask(__name__)
22     app.config['SECRET_KEY'] = 'secretpassword'
23     bootstrap = Bootstrap(app)
24     class TitanicForm(FlaskForm):
25         name = StringField('Name')
26         sex = RadioField('Gender:', coerce=str, 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'),('6-80-88')])
28         ticket = StringField('Ticket:')
29         fare = FloatField('Fare')
30         alive = RadioField('Survived')
31         parch = IntegerField('Number of parents/children')
32         sibsp = IntegerField('Number of siblings/spouses')
33         embarked = RadioField('Embark 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             mint (form)

```

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 ellipse 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:	
<input checked="" type="radio"/>	First
<input type="radio"/>	Second
<input type="radio"/>	Third
Name: Bernie Beekman	
Gender:	
<input checked="" type="radio"/>	Male
<input type="radio"/>	Female
Number of siblings/spouses: 1	
Number of parents/children: 1	
Ticket: 1234	
Fare: 23	
Embark Location:	
<input checked="" type="radio"/>	South Hampton
<input type="radio"/>	Cherbourg
<input type="radio"/>	Queenstown
Age:	
<input checked="" type="radio"/>	0-5
<input type="radio"/>	6-11
<input type="radio"/>	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!](#)