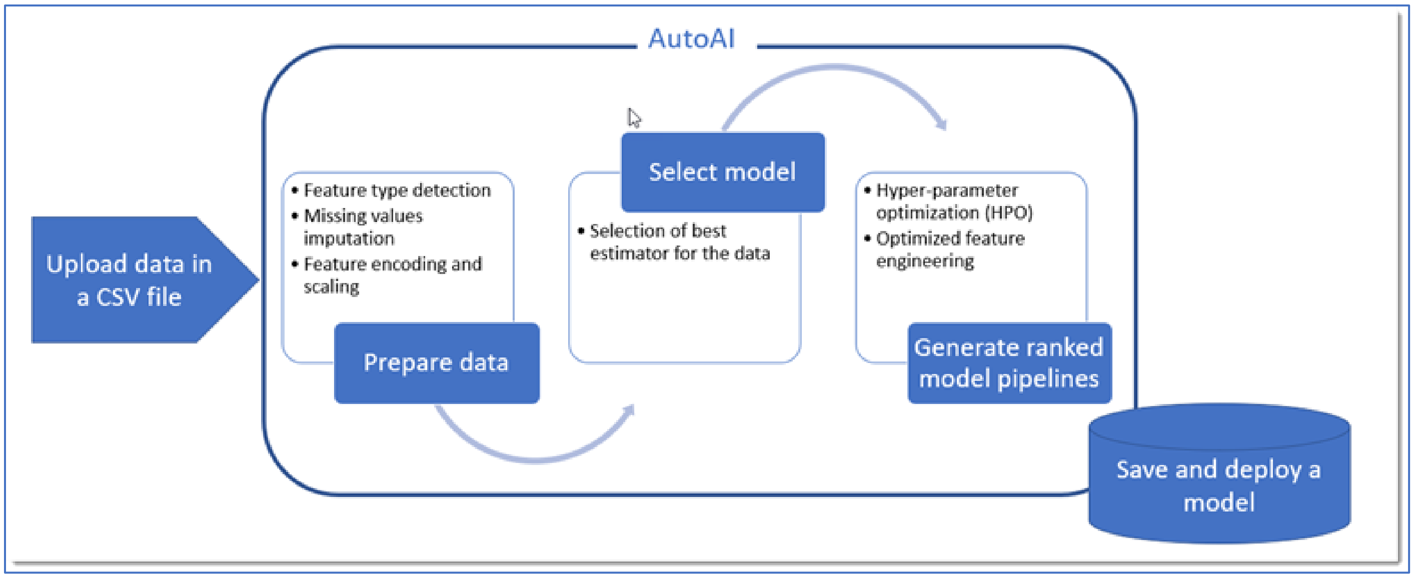
AutoAI Lab + DevOps

This lab consists of two parts. The first part will demonstrate the new and exciting AutoAI capability to build and deploy an optimized model based on the Titanic data set. The second part will deploy an application using the IBM Cloud DevOps toolchain that will invoke the deployed model to predict whether a passenger would have survived.

AutoAI in Watson Studio automatically analyzes your data and generates candidate model pipelines customized for your predictive modeling problem. AutoAI algorithms analyze your dataset to discover data transformations, estimator algorithms, and parameter settings that work best for your problem setting. Results are displayed on a leaderboard, showing the automatically generated model pipelines ranked according to your problem optimization objective.

Using AutoAI, you can build and deploy a machine learning model with sophisticated training features and no coding. The tool does most of the work for you.


AutoAI
Upload data in a CSV file
• Feature type detection
• Missing values imputation
• Feature encoding and scaling
Prepare data
Select model
• Selection of best estimator for the data
• Hyper-parameter optimization (HPO)
• Optimized feature engineering
Generate ranked model pipelines
Save and deploy a model

The AutoAI process follows this sequence to build candidate pipelines:

- Data pre-processing
- Automated model selection
- Automated feature engineering
- Hyperparameter optimization

We will perform the following steps in this lab:

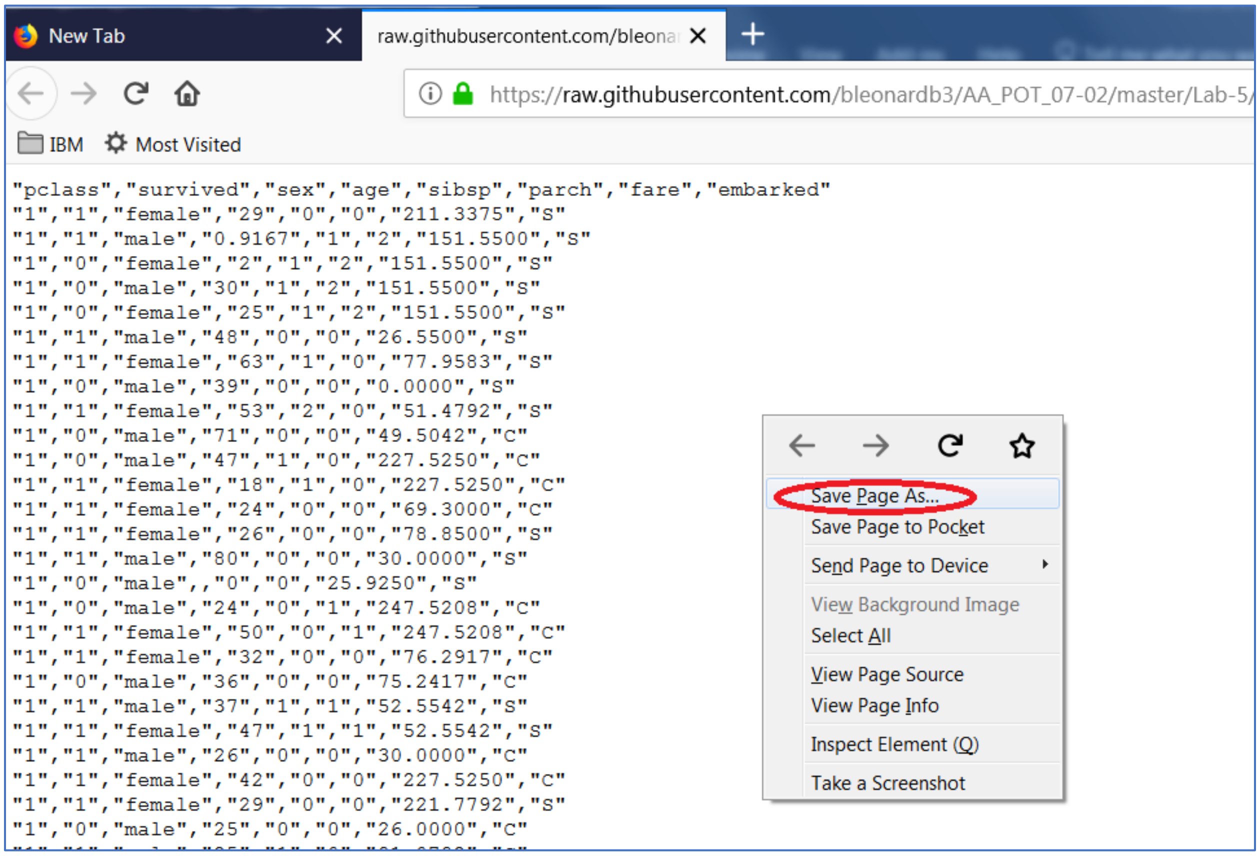
1. Download the Titanic data set
2. Add an Auto AI Experiment
3. Save and Deploy the selected model.

4. Deploying a simple web front-end application and connecting it to the deployed model using an IBM Cloud toolchain.

Step 1: Download the titanic_cleansed.csv data set

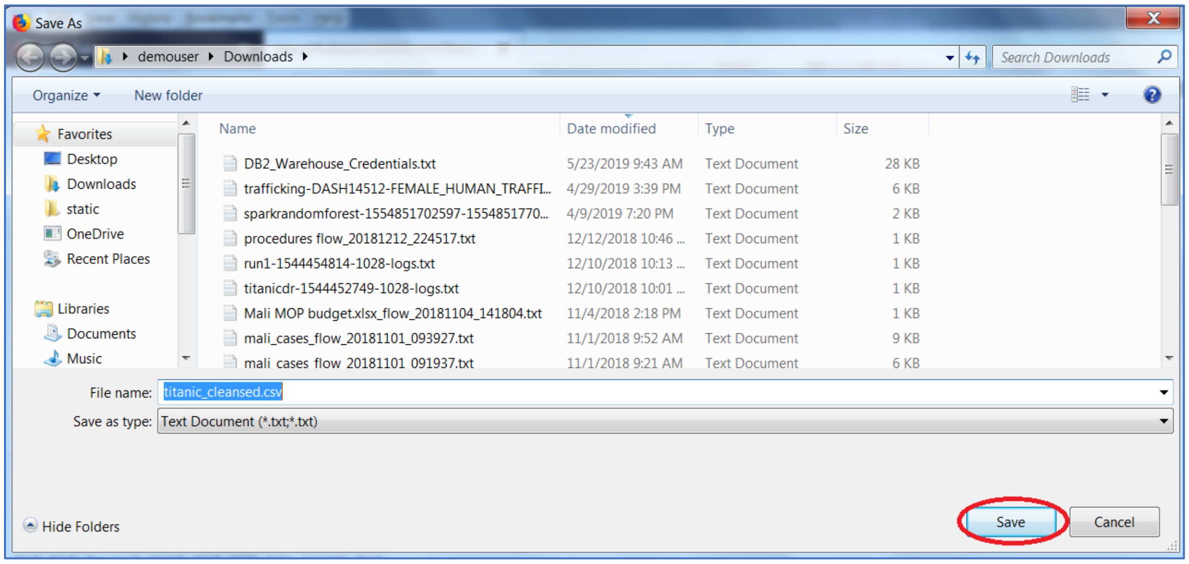
1. Download the titanic_cleansed.csv data file from the following location by clicking on the link Titanic Data.

2. Right-click on the window, and click **Save Page As…**



```
"pclass","survived","sex","age","sibsp","parch","fare","embarked"
"1","1","female","29","0","0","211.3375","S"
"1","1","male","0.9167","1","2","151.5500","S"
"1","0","female","2","1","2","151.5500","S"
"1","0","male","30","1","2","151.5500","S"
"1","0","female","25","1","2","151.5500","S"
"1","1","male","48","0","0","26.5500","S"
"1","1","female","63","1","0","77.9583","S"
"1","0","male","39","0","0","0.0000","S"
"1","1","female","53","2","0","51.4792","S"
"1","0","male","71","0","0","49.5042","C"
"1","0","male","47","1","0","227.5250","C"
"1","1","female","18","1","0","227.5250","C"
"1","1","female","24","0","0","69.3000","C"
"1","1","female","26","0","0","78.8500","S"
"1","1","male","80","0","0","30.0000","S"
"1","0","male",,"0","0","25.9250","S"
"1","0","male","24","0","1","247.5208","C"
"1","1","female","50","0","1","247.5208","C"
"1","1","female","32","0","0","76.2917","C"
"1","0","male","36","0","0","75.2417","C"
"1","1","male","37","1","1","52.5542","S"
"1","1","female","47","1","1","52.5542","S"
"1","1","male","26","0","0","30.0000","C"
"1","1","female","42","0","0","227.5250","C"
"1","1","female","29","0","0","221.7792","S"
"1","0","male","25","0","0","26.0000","C"
```

3. Click on **Save**.


Save As

demouser ▸ Downloads ▸

Search Downloads

Organize ▾ New folder

Favorites
Desktop
Downloads
static
OneDrive
Recent Places

Libraries
Documents
Music

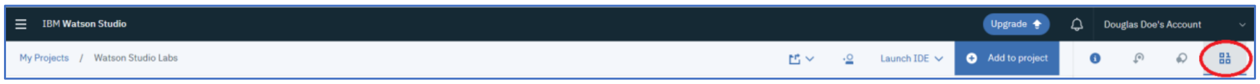
Name	Date modified	Type	Size
DB2_Warehouse_Credentials.txt	5/23/2019 9:43 AM	Text Document	28 KB
trafficking-DASH14512-FEMALE_HUMAN_TRAFFI...	4/29/2019 3:39 PM	Text Document	6 KB
sparkrandomforest-1554851702597-1554851770...	4/9/2019 7:20 PM	Text Document	2 KB
procedures flow_20181212_224517.txt	12/12/2018 10:46 ...	Text Document	1 KB
run1-1544454814-1028-logs.txt	12/10/2018 10:13 ...	Text Document	1 KB
titanicdr-1544452749-1028-logs.txt	12/10/2018 10:01 ...	Text Document	1 KB
Mali MOP budget.xlsx_flow_20181104_141804.txt	11/4/2018 2:18 PM	Text Document	1 KB
mali_cases_flow_20181101_093927.txt	11/1/2018 9:52 AM	Text Document	9 KB
mali_cases_flow_20181101_091937.txt	11/1/2018 9:21 AM	Text Document	6 KB

File name: titanic_cleansed.csv

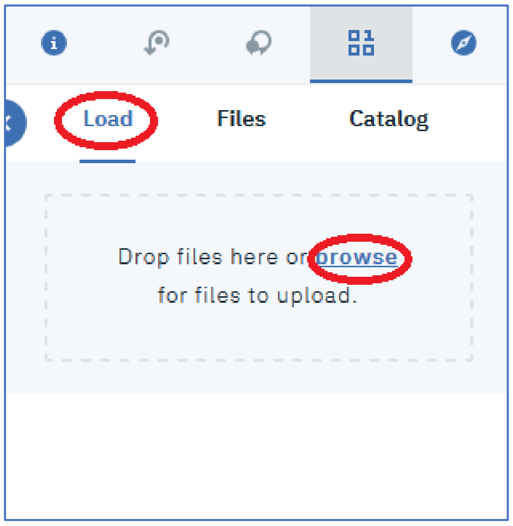
Save as type: Text Document (*.txt;*.txt)

Hide Folders

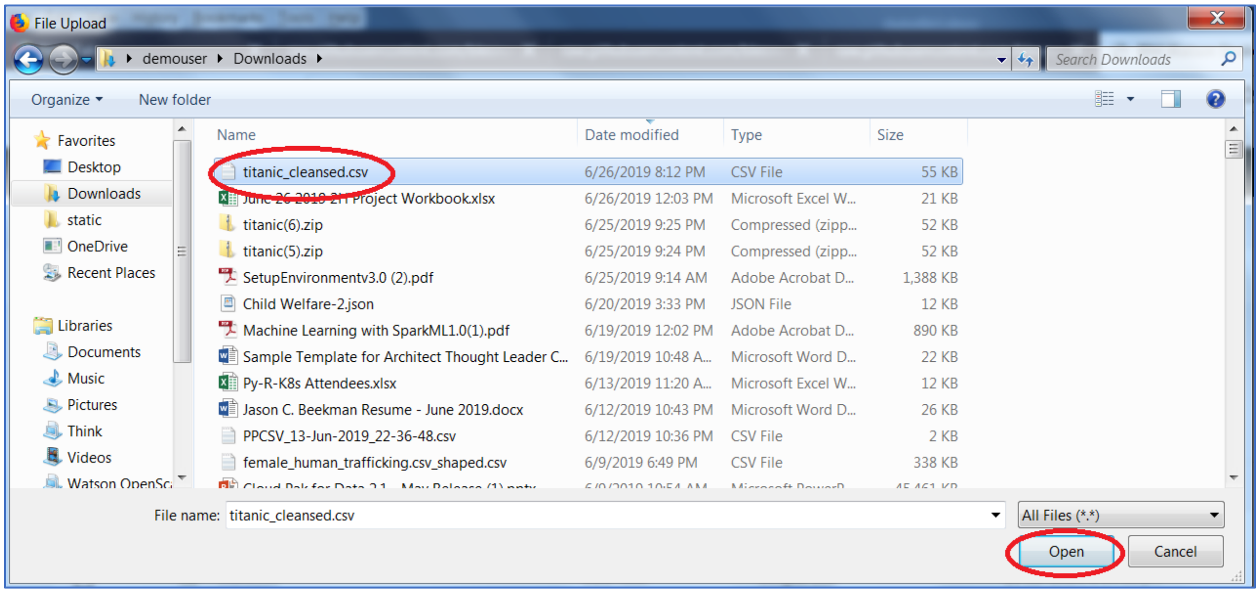
4. Go back to your Watson Studio Labs project. Click on the icon.



5. Click on the **Load** tab and then click on **browse**.



6. Go to the folder where the titanic_cleansed.csv file is stored. Select the titanic_cleansed.csv file and then click **Open**.



File Upload

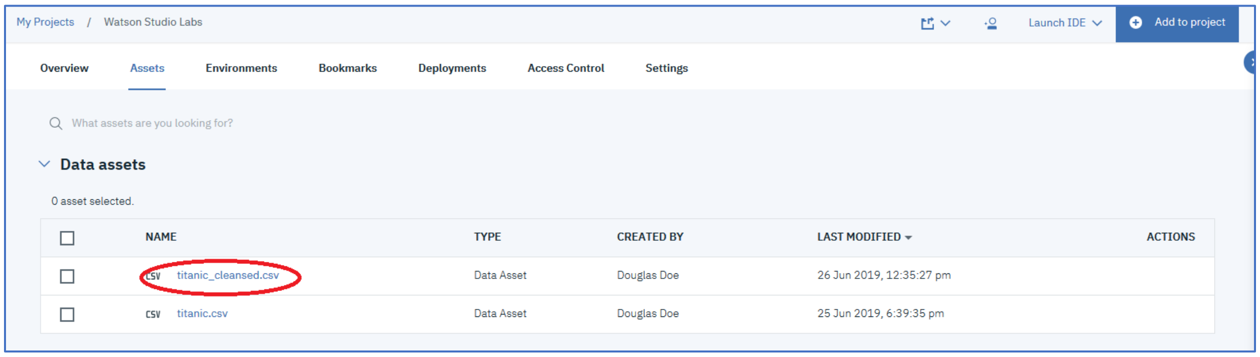
demouser ▸ Downloads ▸

Name	Date modified	Type	Size
titanic_cleansed.csv	6/26/2019 8:12 PM	CSV File	55 KB
June-26-2019-211 Project Workbook.xlsx	6/26/2019 12:03 PM	Microsoft Excel W...	21 KB
titanic(6).zip	6/25/2019 9:25 PM	Compressed (zipp...	52 KB
titanic(5).zip	6/25/2019 9:24 PM	Compressed (zipp...	52 KB
SetupEnvironmentv3.0 (2).pdf	6/25/2019 9:14 AM	Adobe Acrobat D...	1,388 KB
Child Welfare-2.json	6/20/2019 3:33 PM	JSON File	12 KB
Machine Learning with SparkML1.0(1).pdf	6/19/2019 12:02 PM	Adobe Acrobat D...	890 KB
Sample Template for Architect Thought Leader C...	6/19/2019 10:48 A...	Microsoft Word D...	22 KB
Py-R-K8s Attendees.xlsx	6/13/2019 11:20 A...	Microsoft Excel W...	12 KB
Jason C. Beekman Resume - June 2019.docx	6/12/2019 10:43 PM	Microsoft Word D...	26 KB
PPCSV_13-Jun-2019_22-36-48.csv	6/12/2019 10:36 PM	CSV File	2 KB
female_human_trafficking.csv_shaped.csv	6/9/2019 6:49 PM	CSV File	338 KB

File name: titanic_cleansed.csv

All Files (*.*)

7. The titanic_cleansed.csv file is now added as a Data Asset.



My Projects / Watson Studio Labs

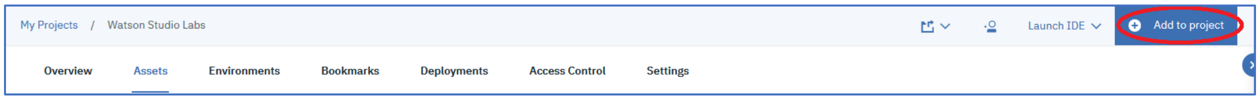
Data assets

0 asset selected.

NAME	TYPE	CREATED BY	LAST MODIFIED	ACTIONS
titanic_cleansed.csv	Data Asset	Douglas Doe	26 Jun 2019, 12:35:27 pm	
titanic.csv	Data Asset	Douglas Doe	25 Jun 2019, 6:39:35 pm	

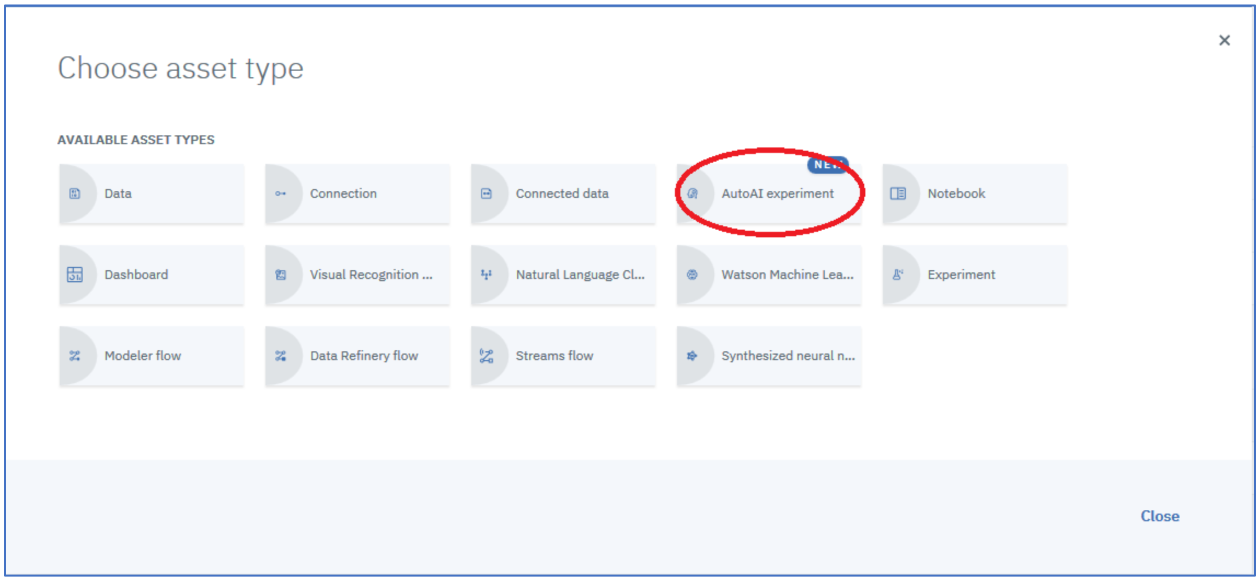
Step 2: Add an AutoAI Experiment

1. Click on **Add to project**.

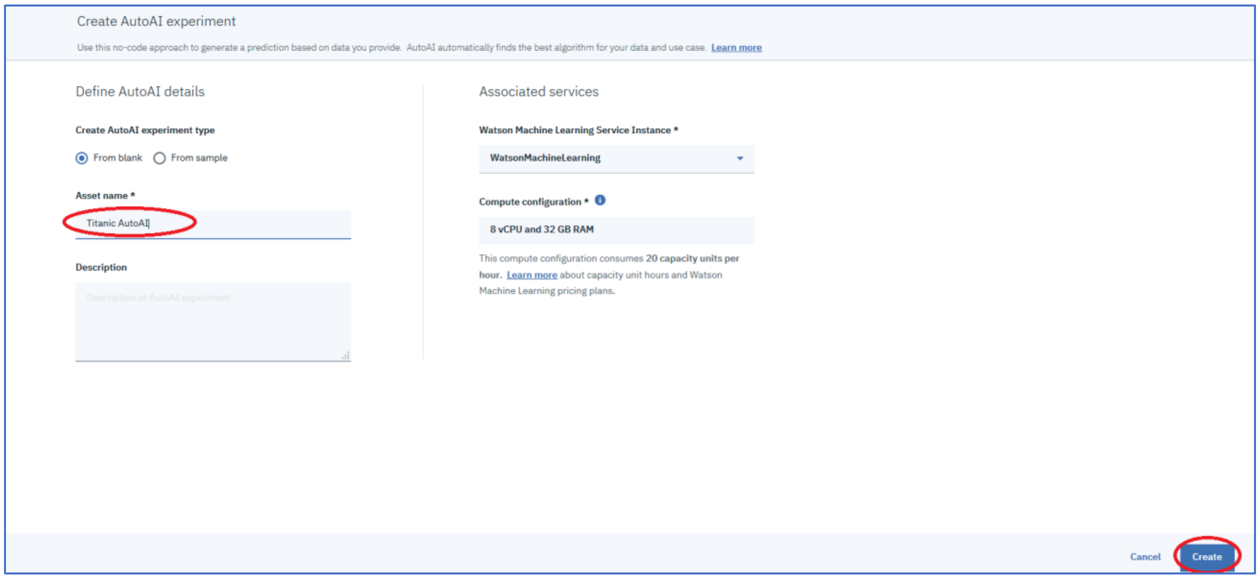


My Projects / Watson Studio Labs

2. Click on **AutoAI experiment**



3. Enter an **Asset name**, leave the defaults for the **Watson Machine Learning** and **Compute configuration** and click on **Create**.



Create AutoAI experiment

Use this no-code approach to generate a prediction based on data you provide. AutoAI automatically finds the best algorithm for your data and use case. Learn more

Define AutoAI details

Create AutoAI experiment type

[✓ From blank] From sample

Asset name *

Titanic AutoAI

Description

Description of AutoAI experiment

Associated services

Watson Machine Learning Service Instance *

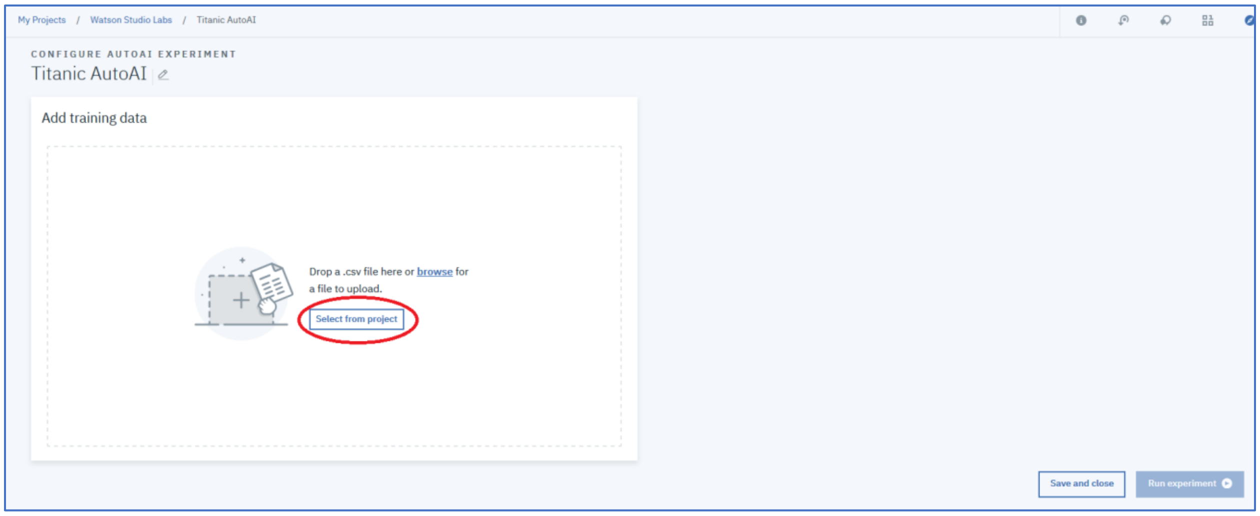
WatsonMachineLearning

Compute configuration *

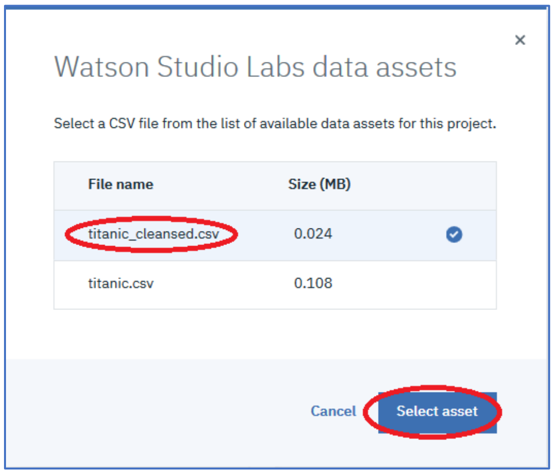
8 vCPU and 32 GB RAM

This compute configuration consumes 20 capacity units per hour. Learn more about capacity unit hours and Watson Machine Learning pricing plans.

4. Click on **Select from project**.


My Projects / Watson Studio Labs / Titanic AutoAI
CONFIGURE AUTOAI EXPERIMENT
Titanic AutoAI
Add training data
Drop a .csv file here or browse for a file to upload.
Select from project
Save and close
Run experiment

5. Click on **titanic_cleansed.csv** and then click on **Select asset**.

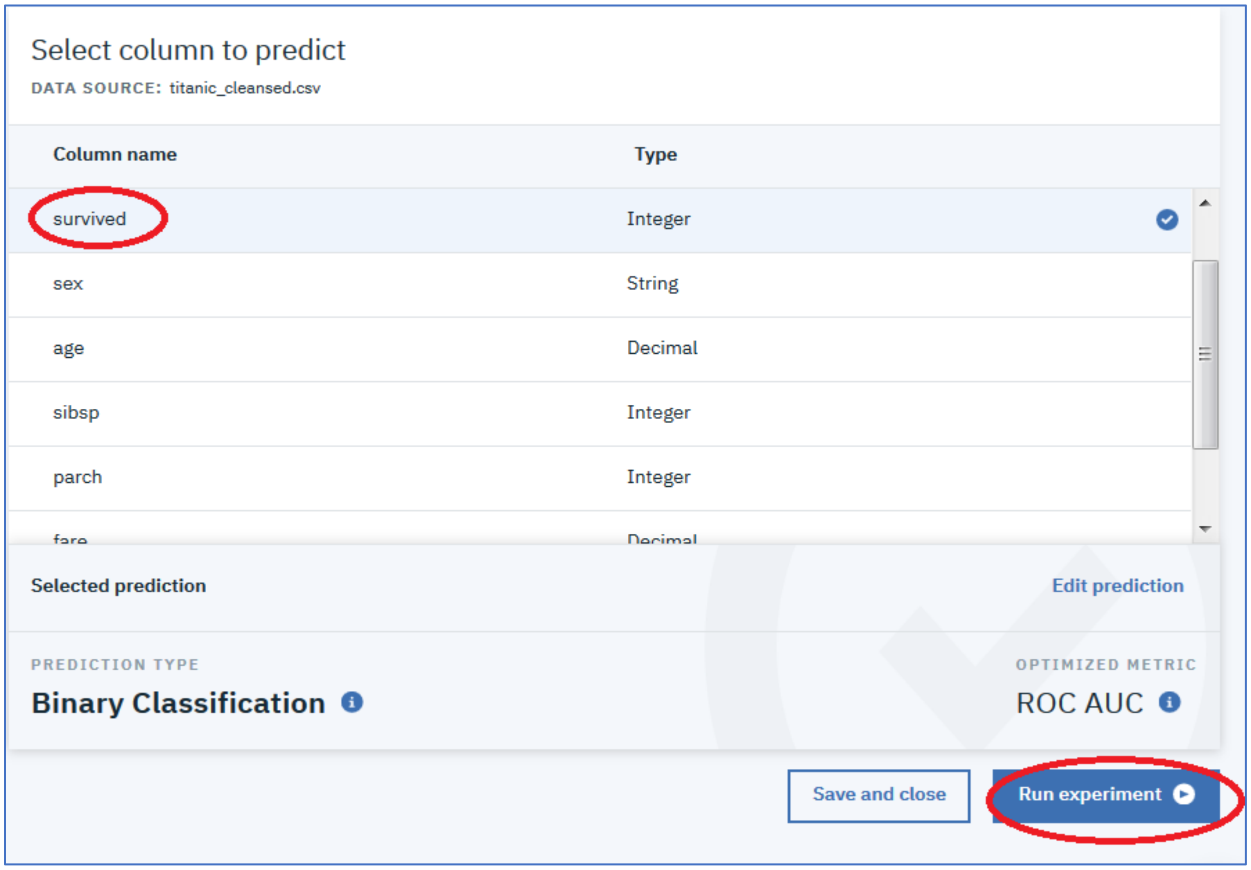


Watson Studio Labs data assets

Select a CSV file from the list of available data assets for this project.

File name	Size (MB)	
titanic_cleansed.csv	0.024	✓
titanic.csv	0.108	

6. Click on **survived** as **the column to predict**, accept the default for **OPTIMIZED METRIC,** and click on **Run experiment.**



Select column to predict

DATA SOURCE: titanic_cleansed.csv

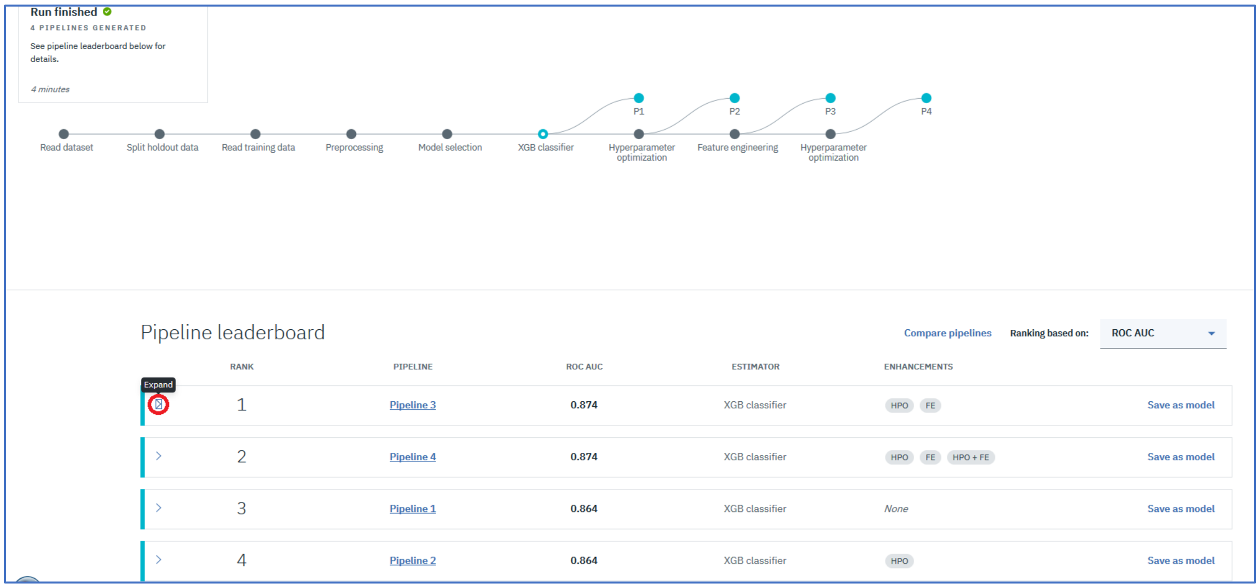
Column name	Type
survived	Integer ✓
sex	String
age	Decimal
sibsp	Integer
parch	Integer
fare	Decimal

Selected prediction — Edit prediction

PREDICTION TYPE: Binary Classification

OPTIMIZED METRIC: ROC AUC

7. The pipeline summary is displayed. Click on the right arrow next to the first pipeline (Pipeline 3).



Run finished

4 PIPELINES GENERATED

See pipeline leaderboard below for details.

4 minutes

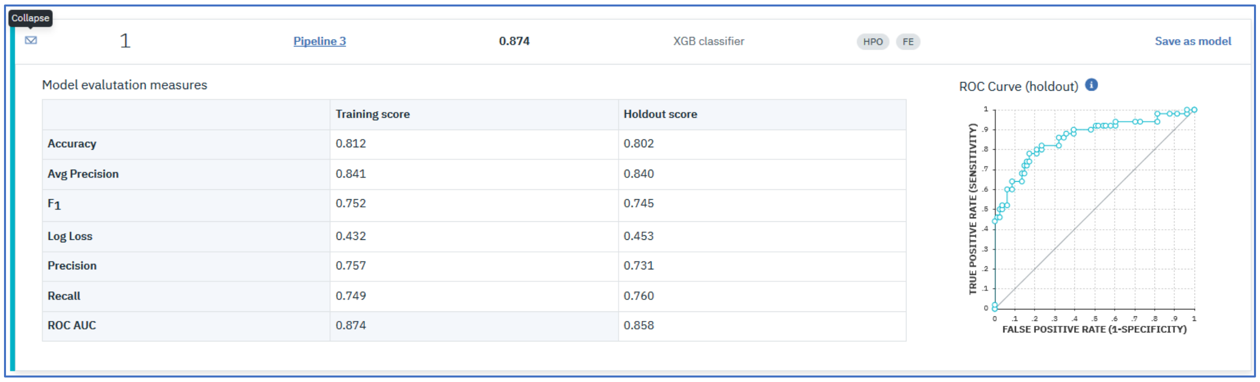
Pipeline leaderboard

Compare pipelines

Ranking based on: ROC AUC

RANK	PIPELINE	ROC AUC	ESTIMATOR	ENHANCEMENTS	
1	Pipeline 3	0.874	XGB classifier	HPO FE	Save as model
2	Pipeline 4	0.874	XGB classifier	HPO FE HPO + FE	Save as model
3	Pipeline 1	0.864	XGB classifier	*None*	Save as model
4	Pipeline 2	0.864	XGB classifier	HPO	Save as model

8. Scores are displayed for different metrics for both the training sample and the holdout sample.



Collapse

1	Pipeline 3	0.874	XGB classifier	HPO FE	Save as model

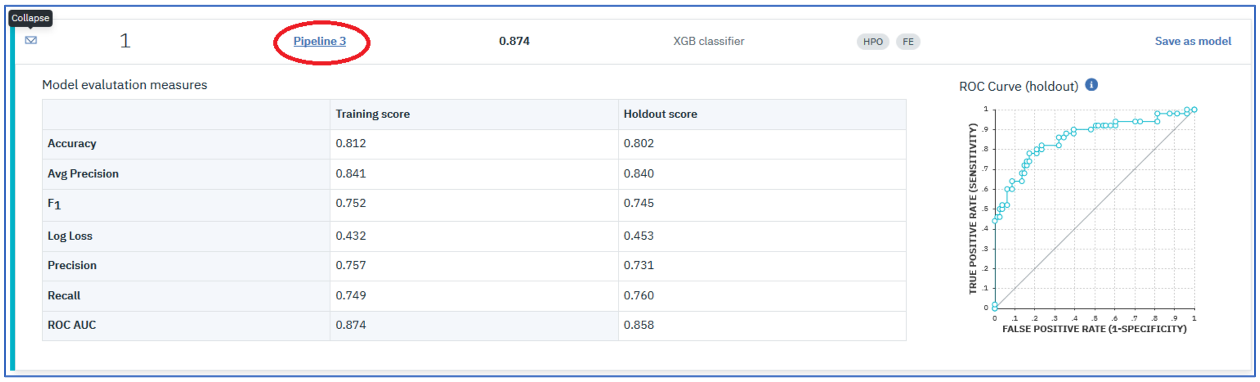
Model evalutation measures

	Training score	Holdout score
Accuracy	0.812	0.802
Avg Precision	0.841	0.840
F1	0.752	0.745
Log Loss	0.432	0.453
Precision	0.757	0.731
Recall	0.749	0.760
ROC AUC	0.874	0.858

ROC Curve (holdout)

TRUE POSITIVE RATE (SENSITIVITY)
FALSE POSITIVE RATE (1-SPECIFICITY)

9. Click on the **Pipeline3** link.



Collapse

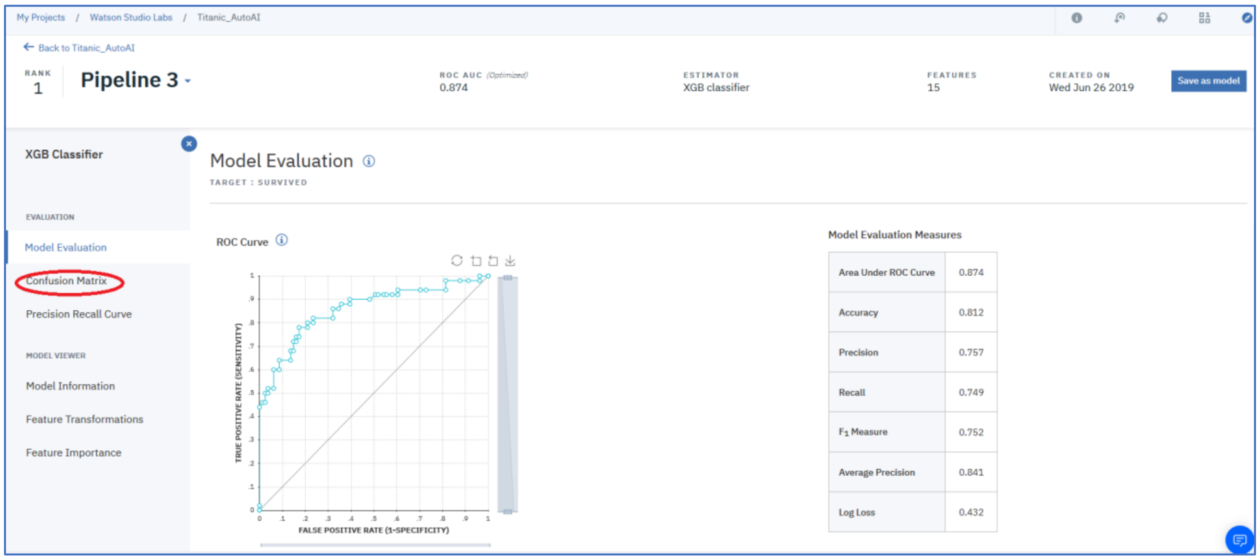
1	Pipeline 3	0.874	XGB classifier	HPO FE	Save as model

Model evalutation measures

	Training score	Holdout score
Accuracy	0.812	0.802
Avg Precision	0.841	0.840
F1	0.752	0.745
Log Loss	0.432	0.453
Precision	0.757	0.731
Recall	0.749	0.760
ROC AUC	0.874	0.858

ROC Curve (holdout)

10. The model evaluation metrics for the training sample are repeated. On the left are options for additional information. Click on the **Confusion Matrix**.



My Projects / Watson Studio Labs / Titanic_AutoAI

Back to Titanic_AutoAI

RANK 1 **Pipeline 3**

ROC AUC (Optimized) 0.874 | ESTIMATOR XGB classifier | FEATURES 15 | CREATED ON Wed Jun 26 2019 | Save as model

Model Evaluation

TARGET : SURVIVED

ROC Curve

TRUE POSITIVE RATE (SENSITIVITY)
FALSE POSITIVE RATE (1-SPECIFICITY)

Model Evaluation Measures

Area Under ROC Curve	0.874
Accuracy	0.812
Precision	0.757
Recall	0.749
F_1 Measure	0.752
Average Precision	0.841
Log Loss	0.432

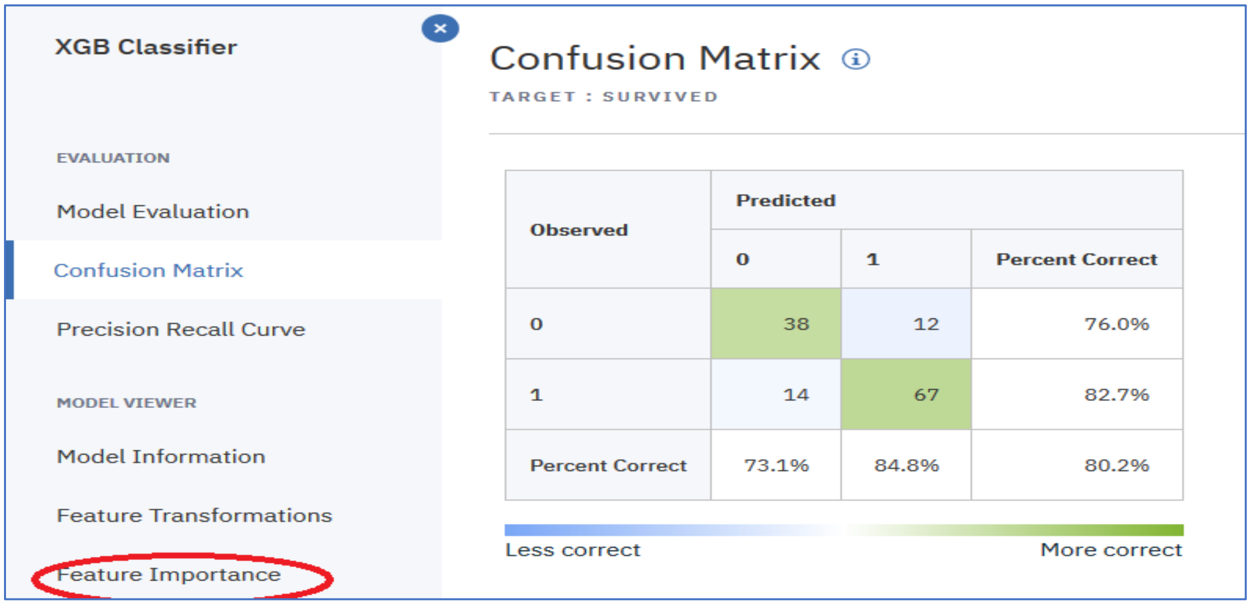
11. The Confusion Matrix is displayed. The different metrics are computed based on the numbers in the Confusion Matrix. For example, Precision is defined by the percentage of predicted positives that are actually positive (i.e. the percentage of predicted survivors that survived). Recall is defined as the percentage of Observed positives that the model

predicts are positive (i.e. the percentage of survivors that the model predicted would survive). Note the higher the Precision the lower the Recall.

Precision = True Positive/ (True Positive + False Positive)

Recall = True Positive/(True Positive + False Negative)

After viewing the Confusion Matrix, click on the **Feature Importance** option.

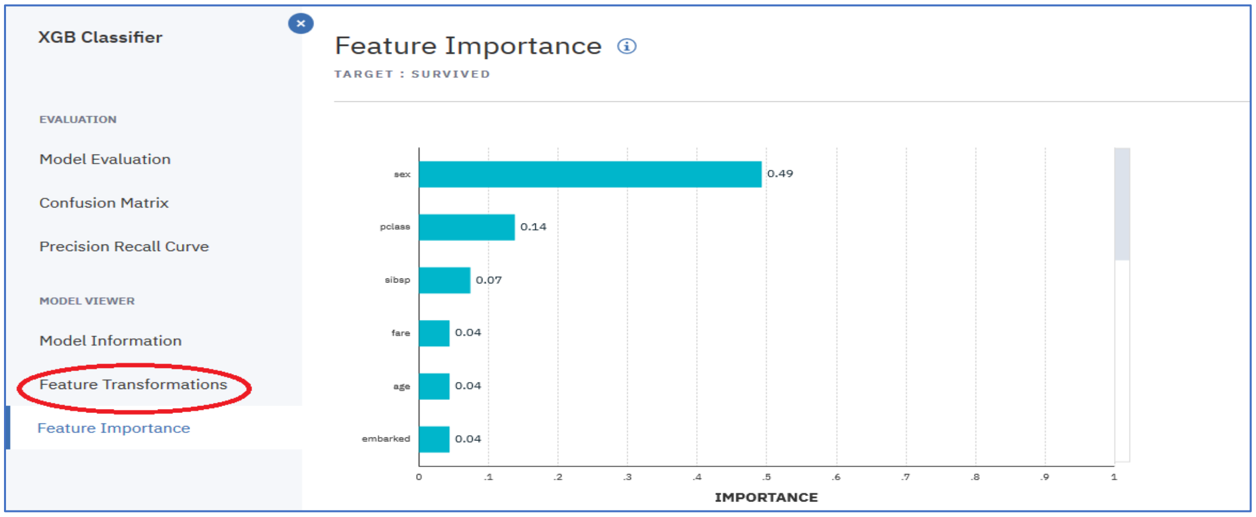


Confusion Matrix

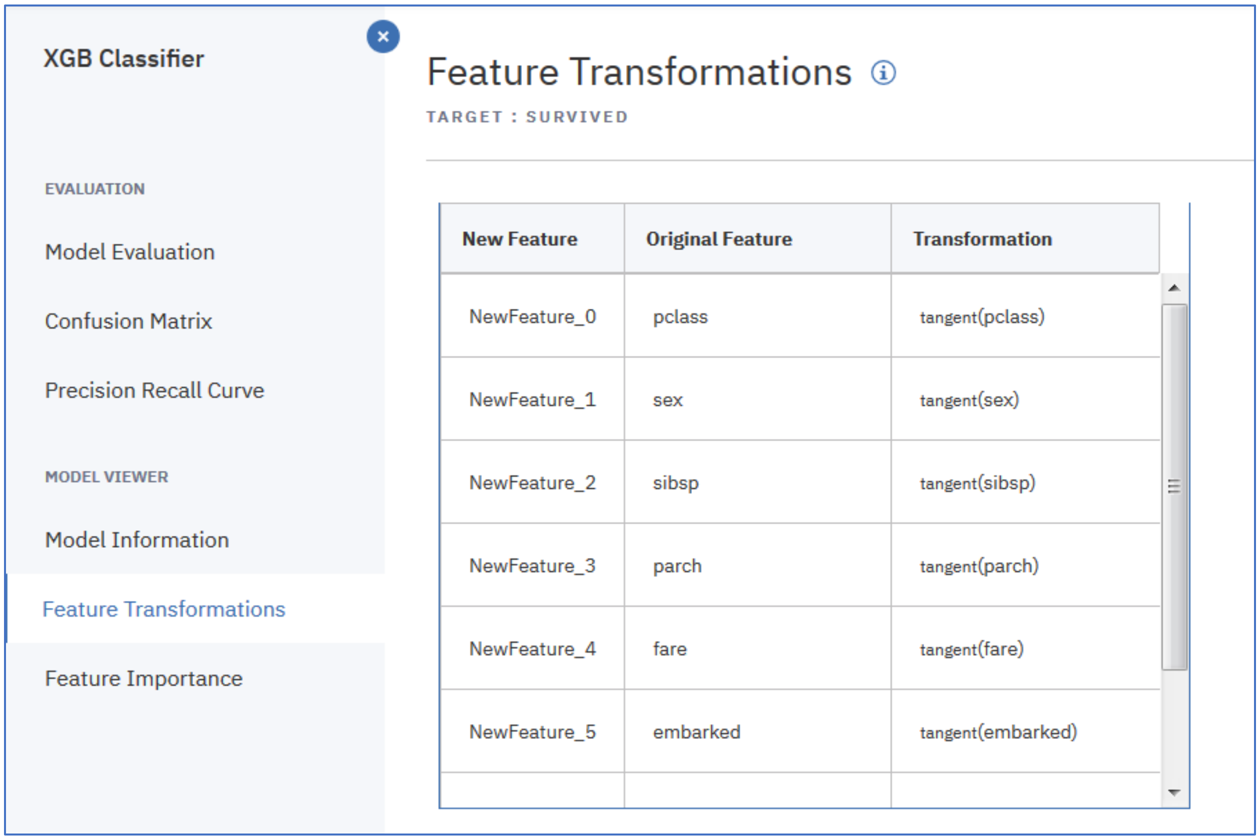
TARGET : SURVIVED

Observed	Predicted		
	0	1	Percent Correct
0	38	12	76.0%
1	14	67	82.7%
Percent Correct	73.1%	84.8%	80.2%

12. According to the Feature Importance, the sex variable is considered the most important feature followed by the passenger class. Click on the **Feature Transformations** option.



13. Feature Transformation applies a transform to features to improve the model performance.



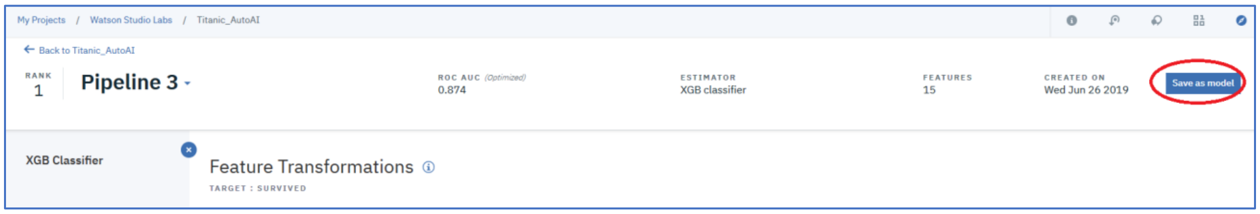
Feature Transformations

TARGET : SURVIVED

New Feature	Original Feature	Transformation
NewFeature_0	pclass	tangent(pclass)
NewFeature_1	sex	tangent(sex)
NewFeature_2	sibsp	tangent(sibsp)
NewFeature_3	parch	tangent(parch)
NewFeature_4	fare	tangent(fare)
NewFeature_5	embarked	tangent(embarked)

Step 3 – Save and Deploy the Selected Model

1. Click on **Save as Model**



RANK 1 Pipeline 3

ROC AUC (Optimized) 0.874

ESTIMATOR XGB classifier

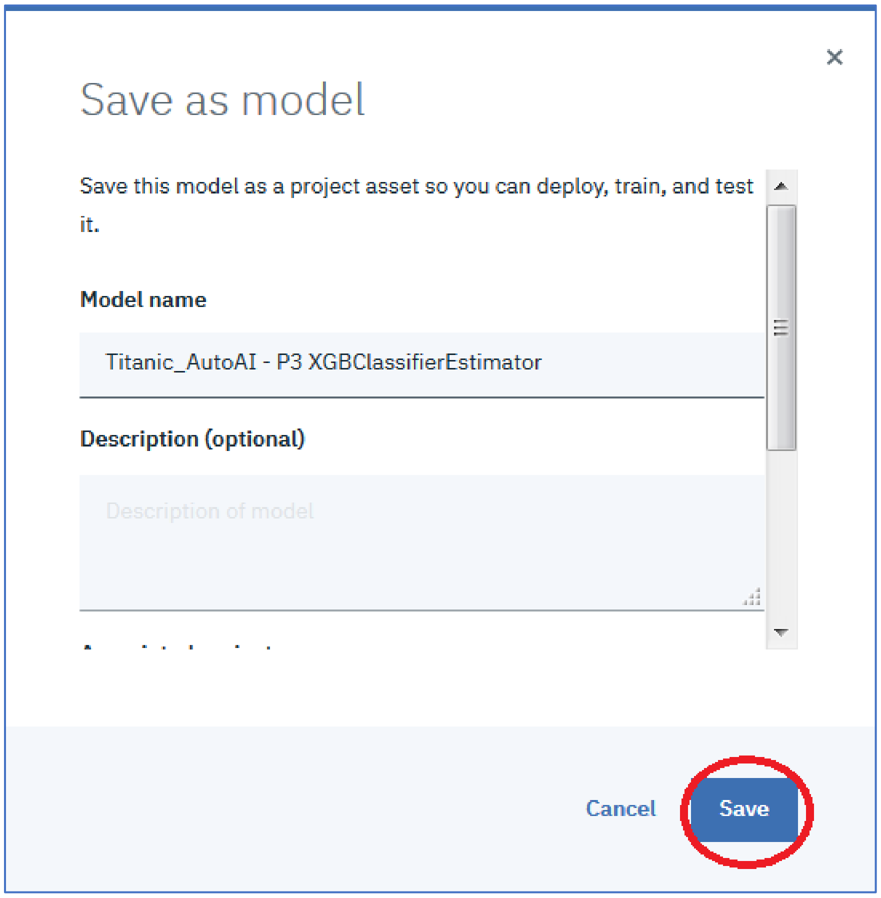
FEATURES 15

CREATED ON Wed Jun 26 2019

Feature Transformations

TARGET : SURVIVED

2. Optionally change the default name and click on **Save**.


Save as model

Save this model as a project asset so you can deploy, train, and test it.

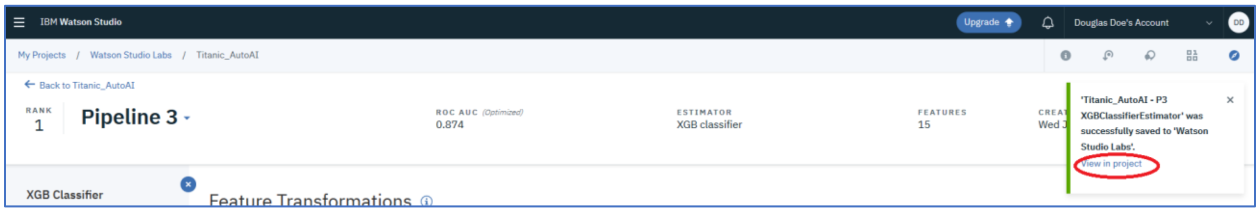
Model name

Titanic_AutoAI - P3 XGBClassifierEstimator

Description (optional)

Description of model

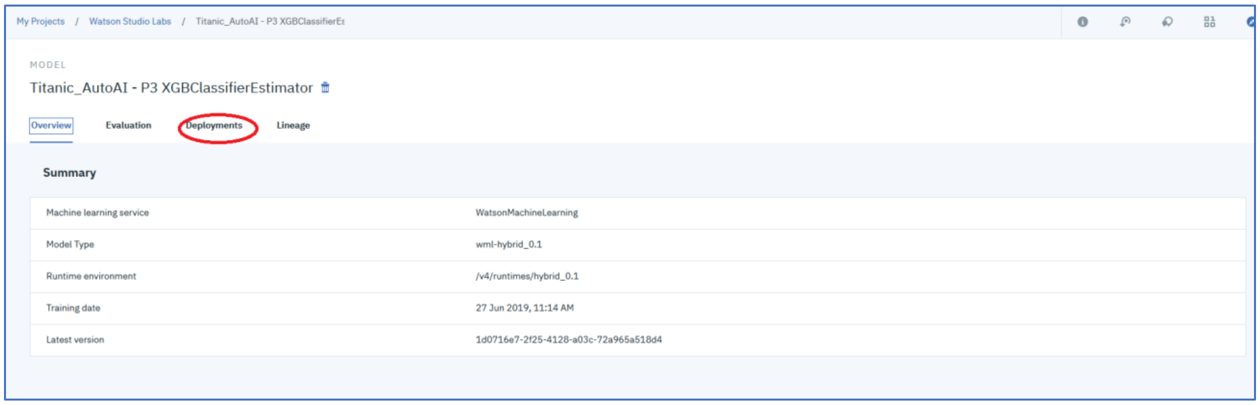
3. Click on **View in Project**.



RANK		ROC AUC *(Optimized)*	ESTIMATOR	FEATURES	CREAT
1	Pipeline 3	0.874	XGB classifier	15	Wed J

'Titanic_AutoAI - P3 XGBClassifierEstimator' was successfully saved to 'Watson Studio Labs'.

4. Click on **Deployments**


My Projects / Watson Studio Labs / Titanic_AutoAI - P3 XGBClassifierEs
MODEL

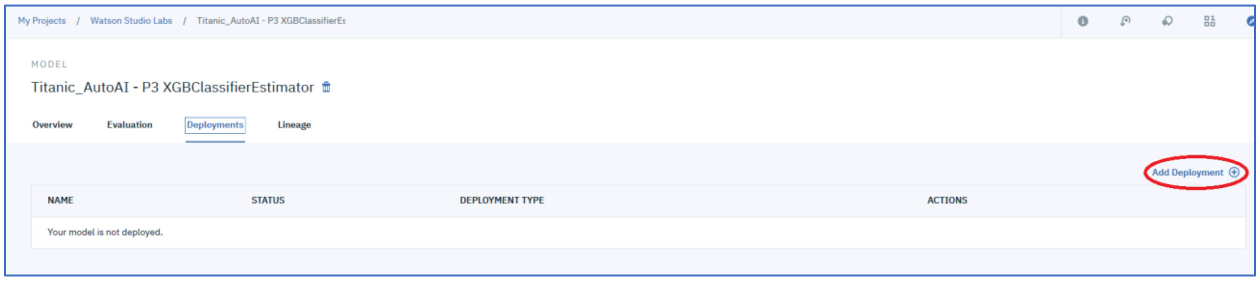
Titanic_AutoAI - P3 XGBClassifierEstimator

Overview
Evaluation
Deployments
Lineage

Summary

Machine learning service	WatsonMachineLearning
Model Type	wml-hybrid_0.1
Runtime environment	/v4/runtimes/hybrid_0.1
Training date	27 Jun 2019, 11:14 AM
Latest version	1d0716e7-2f25-4128-a03c-72a965a518d4

5. Click on **Add Deployment**.



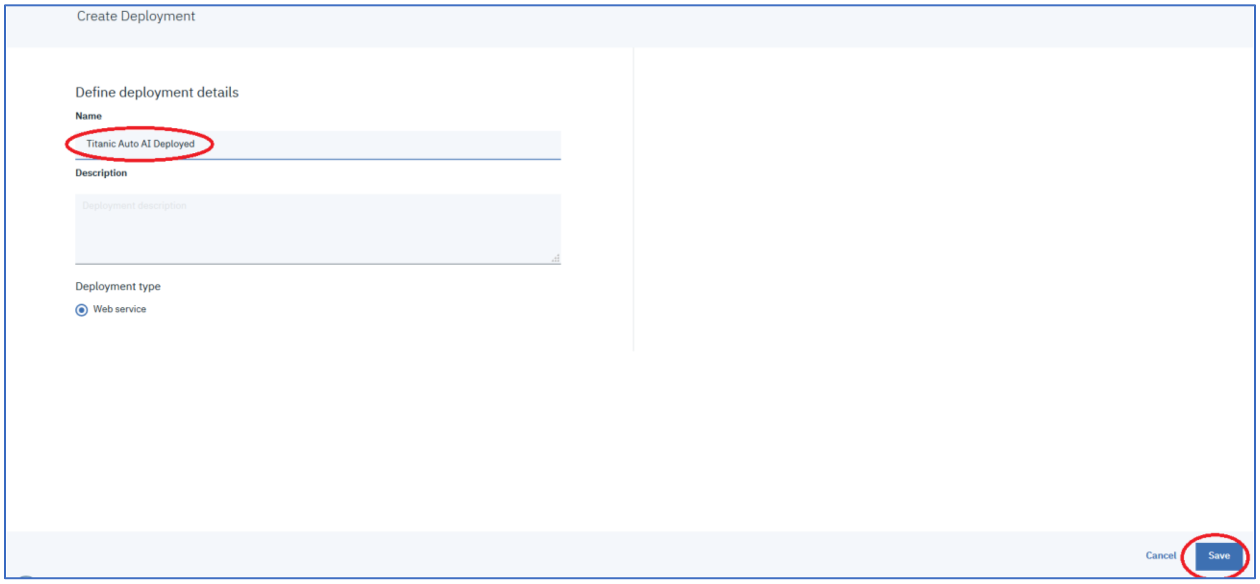
My Projects / Watson Studio Labs / Titanic_AutoAI - P3 XGBClassifierEs

MODEL

Titanic_AutoAI - P3 XGBClassifierEstimator

NAME	STATUS	DEPLOYMENT TYPE	ACTIONS
Your model is not deployed.			

6. Enter a **Name** and click **Save**.



Create Deployment

Define deployment details

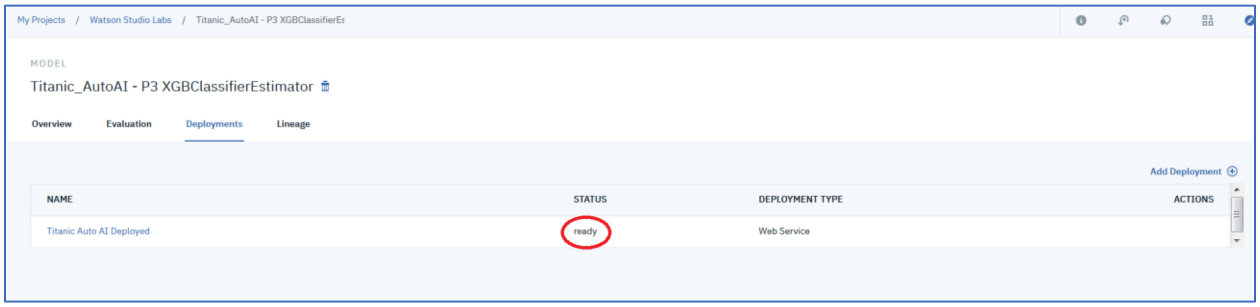
Name: Titanic Auto AI Deployed

Description: Deployment description

Deployment type

Web service

7. The model is successfully deployed on the IBM Cloud.



My Projects / Watson Studio Labs / Titanic_AutoAI - P3 XGBClassifierEs

MODEL

Titanic_AutoAI - P3 XGBClassifierEstimator

NAME	STATUS	DEPLOYMENT TYPE	ACTIONS
Titanic Auto AI Deployed	ready	Web Service	

Step 4: Deploy a simple web front-end to invoke the Watson Machine Learning service

This section will provide an example of a simple Python Flask web front-end application 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.

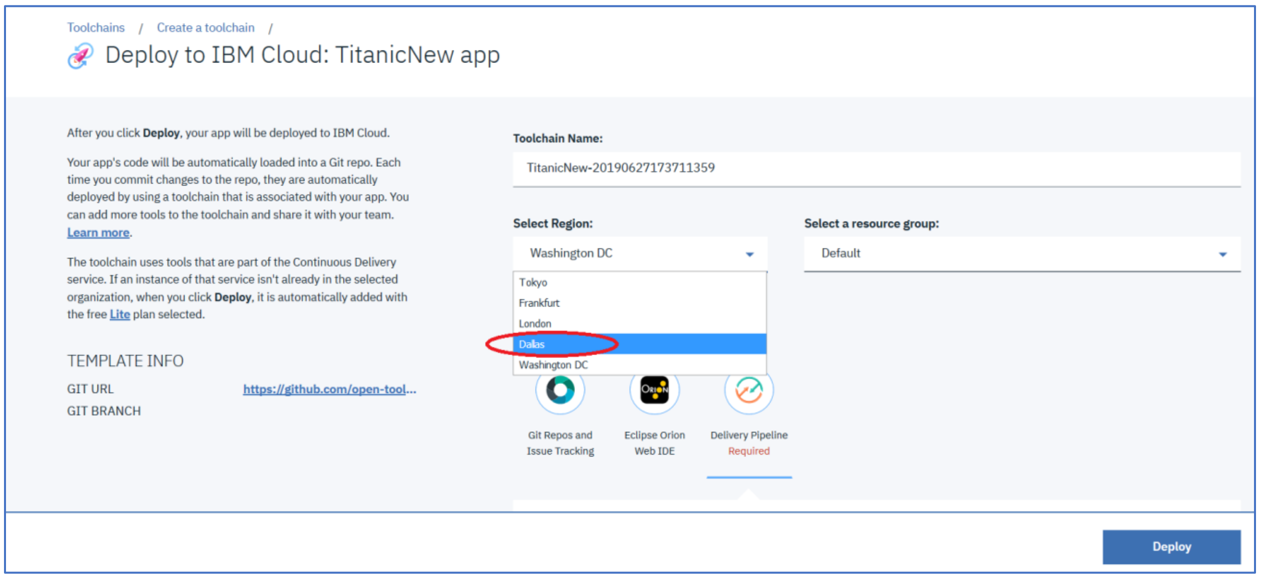
The toolchain uses tools that are part of the Continuous Delivery service. If an instance of that service isn't already in your account, when you click **Deploy**, it is automatically added with the free Lite plan selected.

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

1. Click on the **Deploy to IBM Cloud** link below to deploy a sample Python Flash web application into your IBM Cloud account. Note you may get this message – "*An IBM Cloud account is required. To get started, click Log In or Sign Up at the top of this page*". If you get this message, click on **Log In.**

   Deploy to IBM Cloud

2. In the **Select Region**, make sure that the region is Dallas. If not, change it to Dallas.



Toolchains / Create a toolchain /

Deploy to IBM Cloud: TitanicNew app

After you click **Deploy**, your app will be deployed to IBM Cloud.

Your app's code will be automatically loaded into a Git repo. Each time you commit changes to the repo, they are automatically deployed by using a toolchain that is associated with your app. You can add more tools to the toolchain and share it with your team. Learn more.

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

TEMPLATE INFO

GIT URL https://github.com/open-tool...

GIT BRANCH

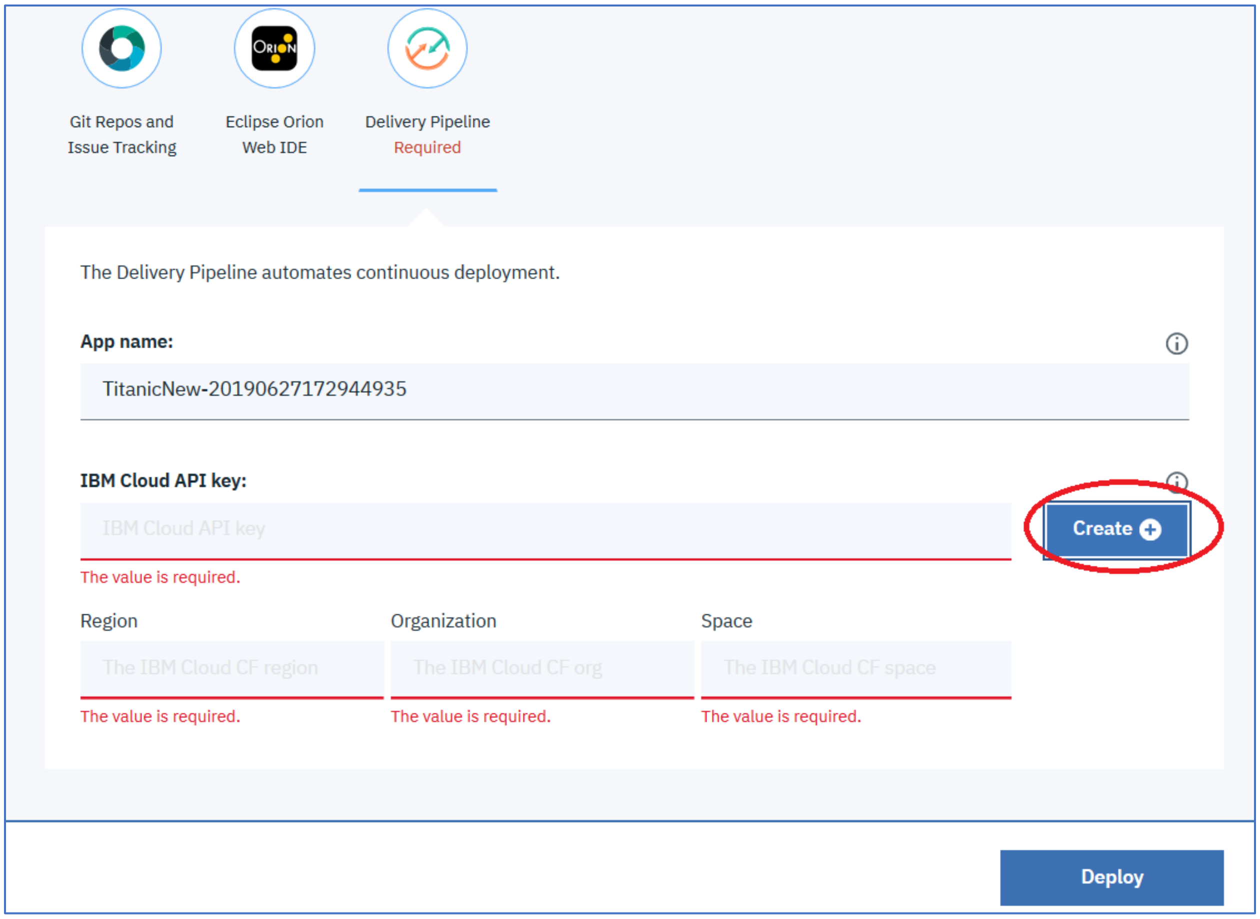
Toolchain Name: TitanicNew-20190627173711359

Select Region: Washington DC

Tokyo
Frankfurt
London
Dallas
Washington DC

Select a resource group: Default

3. Scroll down to the bottom. Click on the **Create+** button to create an IBM Cloud API key.



The Delivery Pipeline automates continuous deployment.

App name:
TitanicNew-20190627172944935

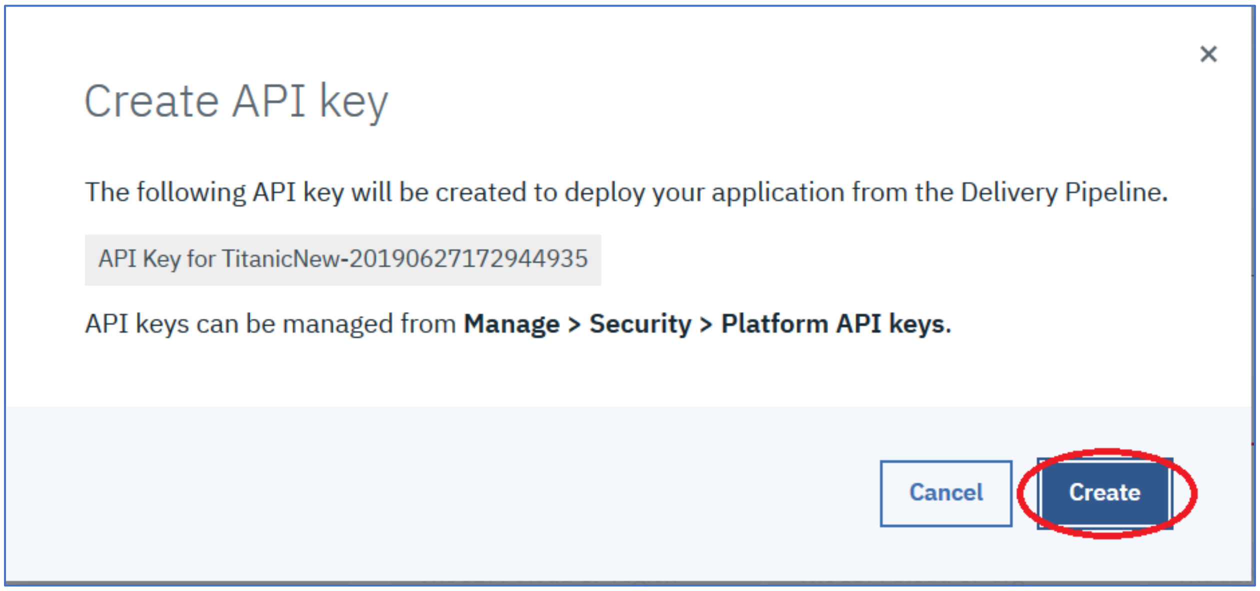
IBM Cloud API key:
IBM Cloud API key
The value is required.

Region
The IBM Cloud CF region
The value is required.

Organization
The IBM Cloud CF org
The value is required.

Space
The IBM Cloud CF space
The value is required.

4. Click on the **Create** button.



Create API key

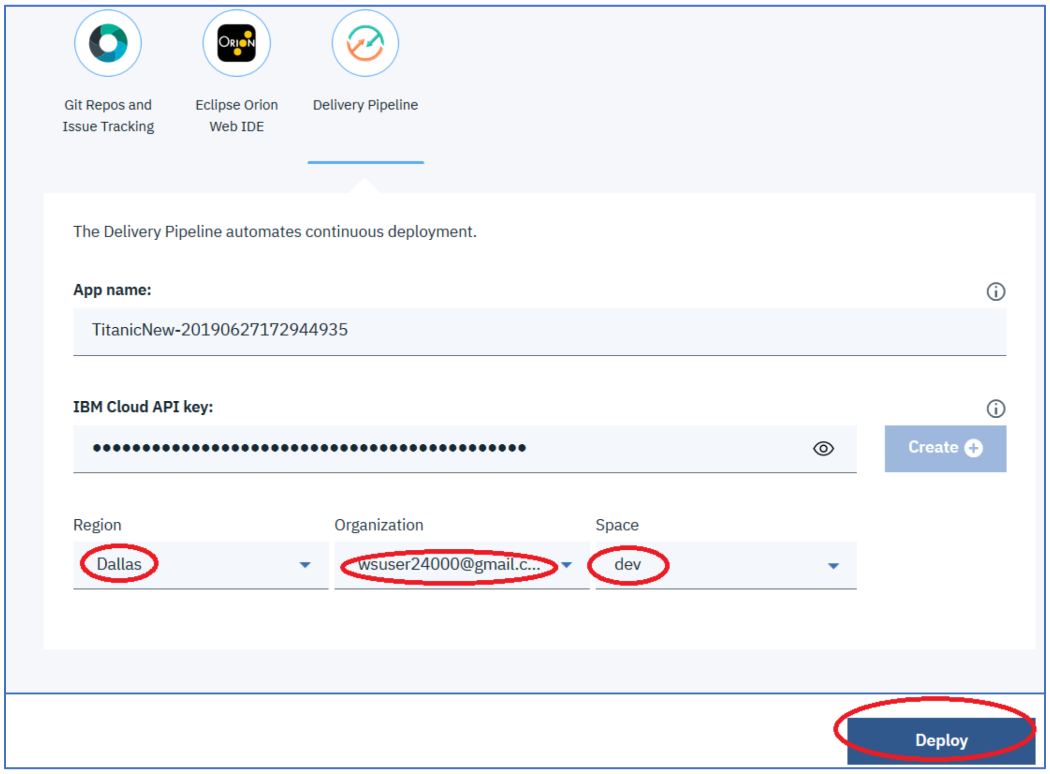
The following API key will be created to deploy your application from the Delivery Pipeline.

API Key for TitanicNew-20190627172944935

API keys can be managed from **Manage > Security > Platform API keys.**

5. Please wait until the Region, Organization, and Space are filled in. **Note that these must match the corresponding Region, Organization, and Space where the Titanic model was deployed.** Make sure the **Region** is Dallas(Production), and the Organization should

display the userid, and the space should be filled in as well. If this is not the case, try a different **Region**. Click on **Deploy**.



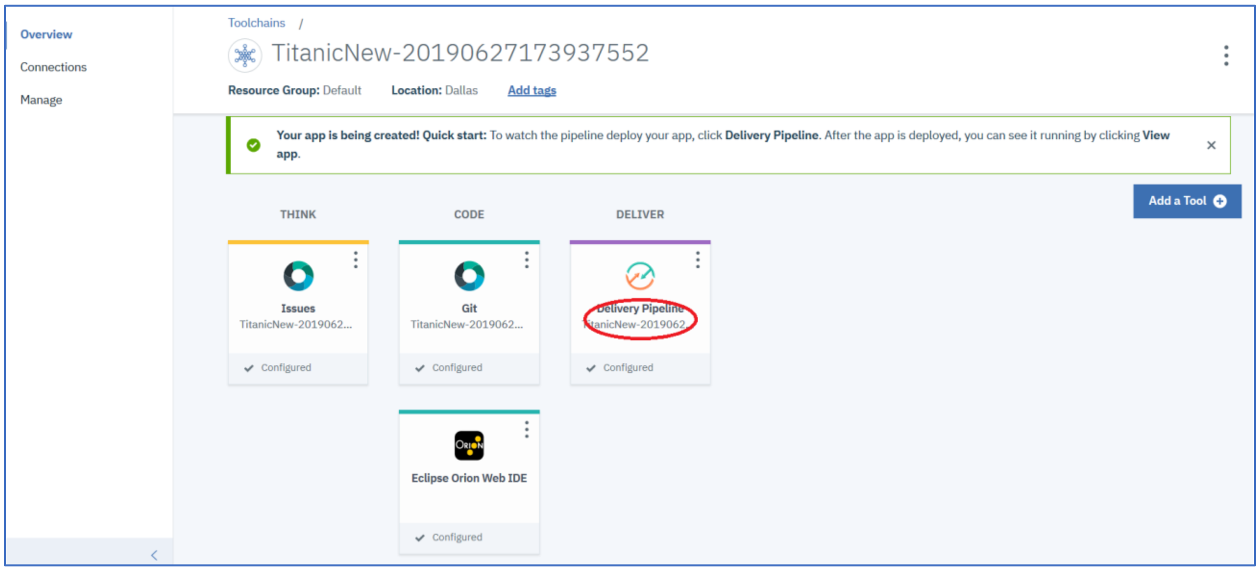
The Delivery Pipeline automates continuous deployment.

App name: TitanicNew-20190627172944935

IBM Cloud API key: ••

Region	Organization	Space
Dallas	wsuser24000@gmail.c...	dev

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

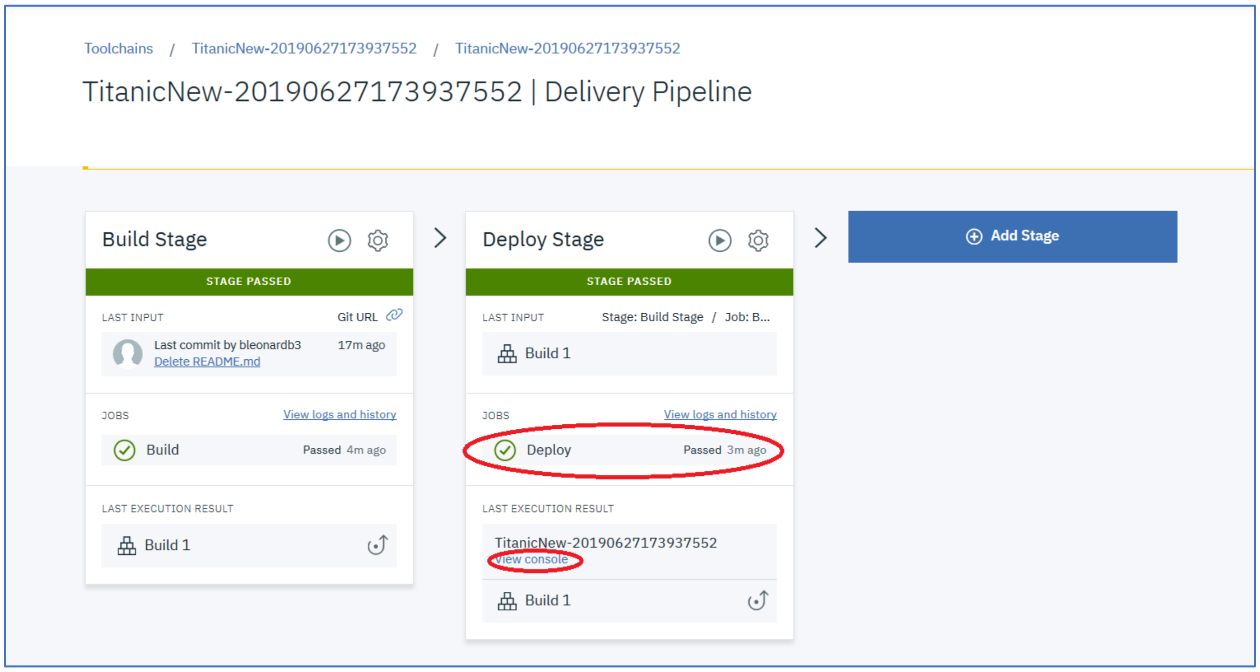


TitanicNew-20190627173937552

Resource Group: Default Location: Dallas Add tags

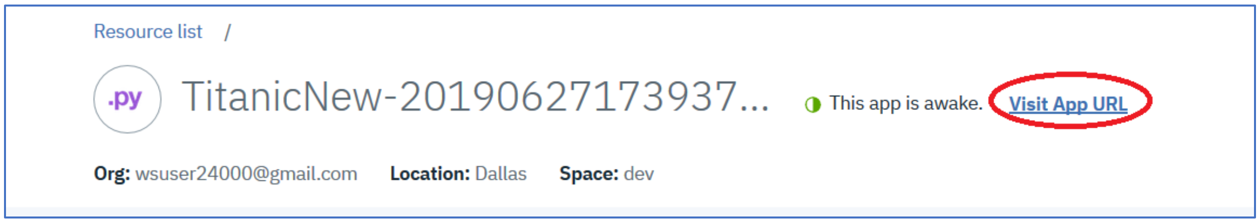
Your app is being created! Quick start: To watch the pipeline deploy your app, click Delivery Pipeline. After the app is deployed, you can see it running by clicking View app.

7. After the app is deployed successfully (should say Deploy Passed in the Deploy stage- may take about 2 minutes), view the running app by clicking on **View Console**


Toolchains / TitanicNew-20190627173937552 / TitanicNew-20190627173937552

TitanicNew-20190627173937552 | Delivery Pipeline

8. Click on **Visit App URL**

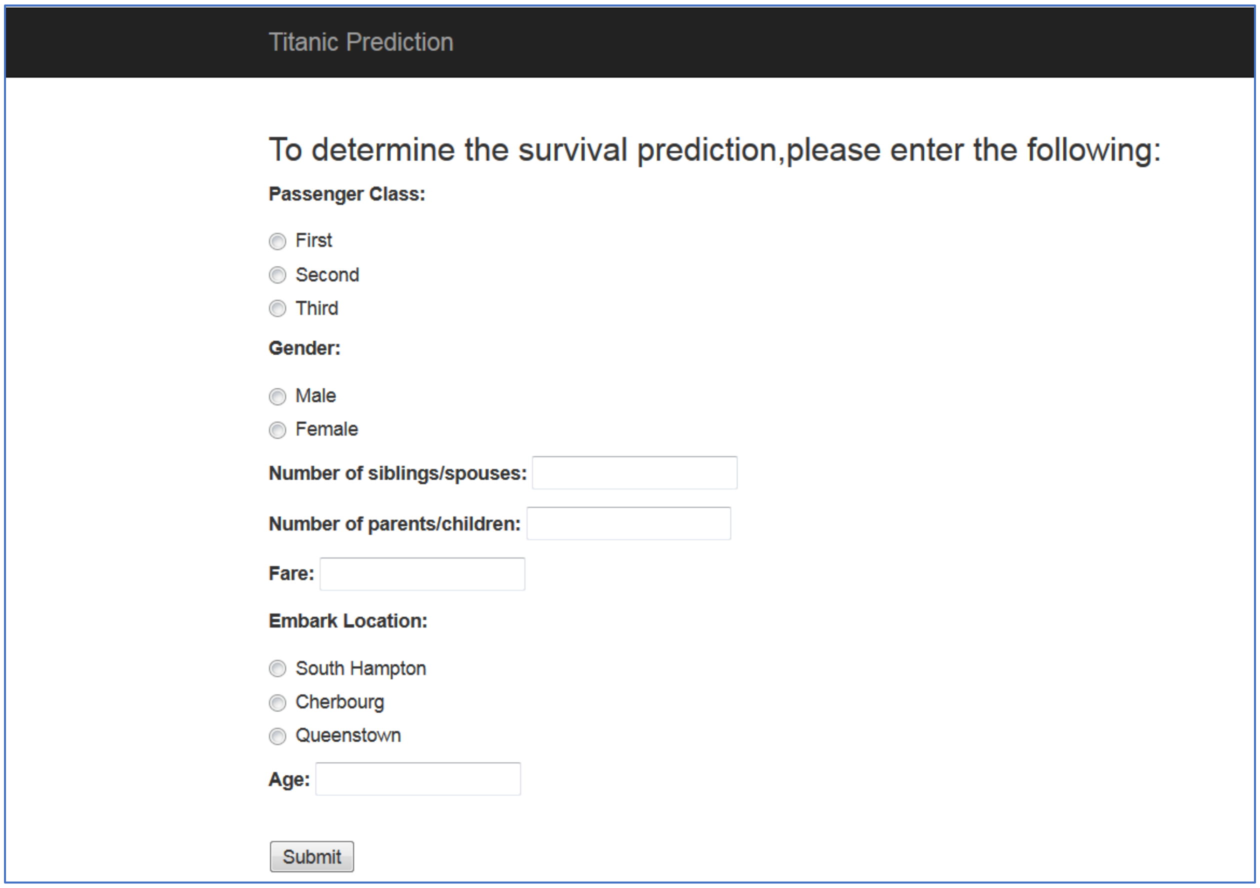

Resource list /

TitanicNew-20190627173937...

This app is awake. Visit App URL

Org: wsuser24000@gmail.com **Location:** Dallas **Space:** dev

9. 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! Close the Titanic Prediction browser tab.



Titanic Prediction

To determine the survival prediction,please enter the following:

Passenger Class:

- First
- Second
- Third

Gender:

- Male
- Female

Number of siblings/spouses:

Number of parents/children:

Fare:

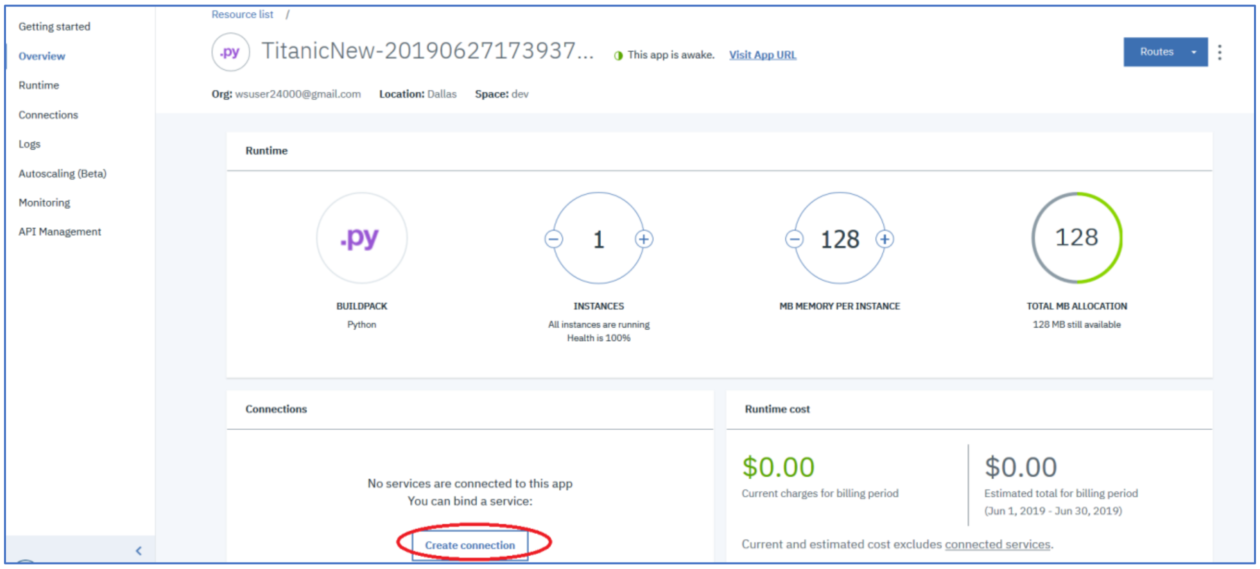
Embark Location:

- South Hampton
- Cherbourg
- Queenstown

Age:

Submit

10. We are now going to connect the application to the Watson Machine Learning service that was created earlier. Scroll down until you see the Connections panel. Click on **Create Connection**.



Getting started
Overview
Runtime
Connections
Logs
Autoscaling (Beta)
Monitoring
API Management

Resource list /

TitanicNew-20190627173937... This app is awake. Visit App URL

Routes

Org: wsuser24000@gmail.com Location: Dallas Space: dev

Runtime

BUILDPACK	INSTANCES	MB MEMORY PER INSTANCE	TOTAL MB ALLOCATION
.py	1	128	128
Python	All instances are running Health is 100%		128 MB still available

Connections

No services are connected to this app
You can bind a service:

Create connection

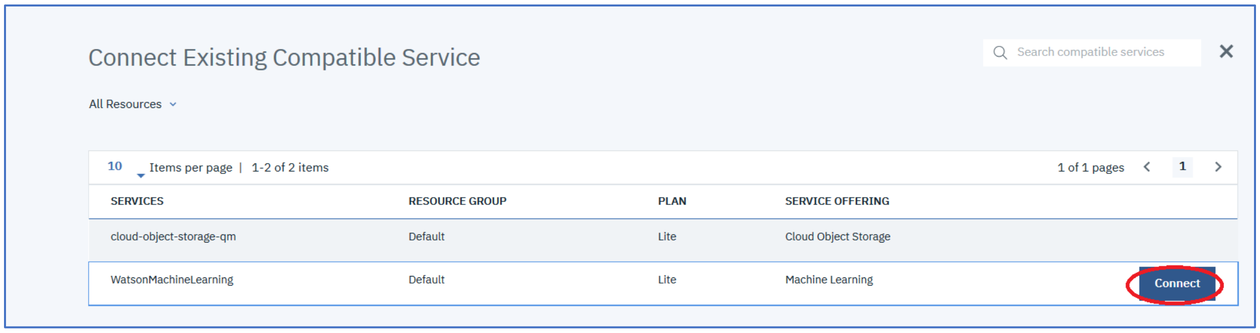
Runtime cost

$0.00
Current charges for billing period

$0.00
Estimated total for billing period
(Jun 1, 2019 - Jun 30, 2019)

Current and estimated cost excludes connected services.

11. You should see at least 2 services listed, a Cloud Object Storage service and a Watson Machine Learning service. Point the cursor on the **Watson Machine Learning** service for your application, and then click on **Connect**.



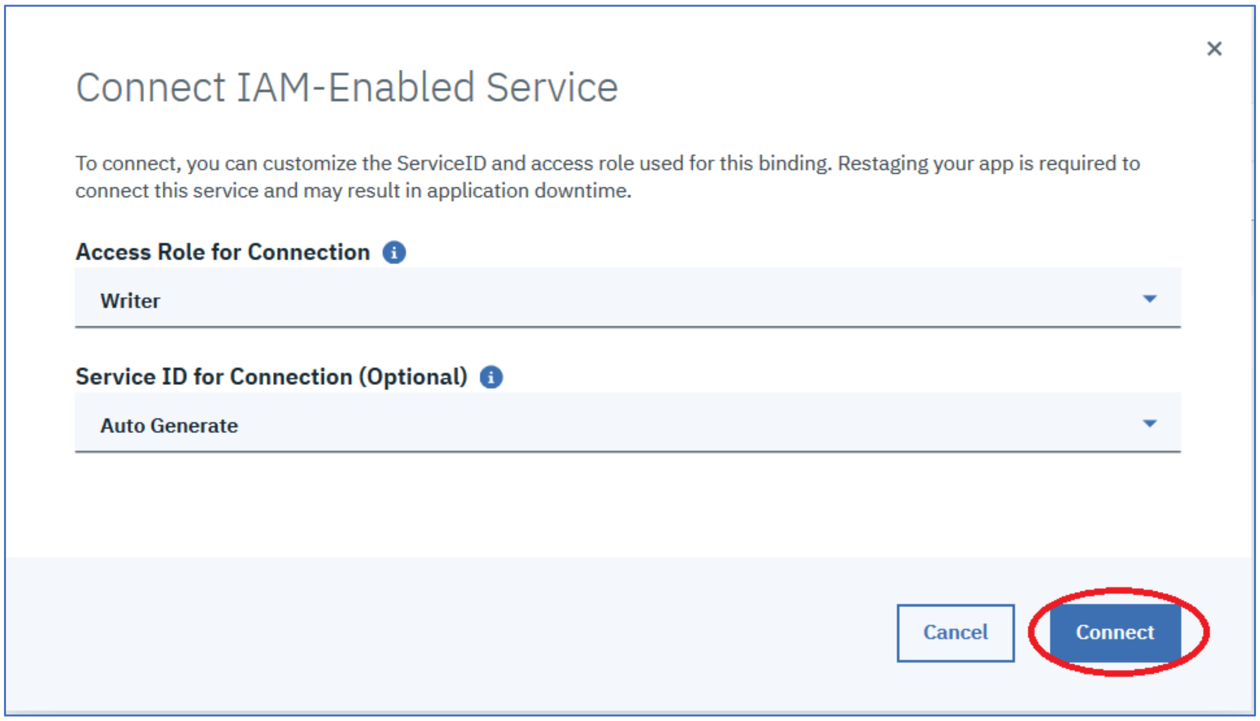
Connect Existing Compatible Service

All Resources

10 Items per page | 1-2 of 2 items — 1 of 1 pages

SERVICES	RESOURCE GROUP	PLAN	SERVICE OFFERING	
cloud-object-storage-qm	Default	Lite	Cloud Object Storage	
WatsonMachineLearning	Default	Lite	Machine Learning	Connect

12. A **Connect IAM-enabled service** pop up will appear. Click **Connect**.



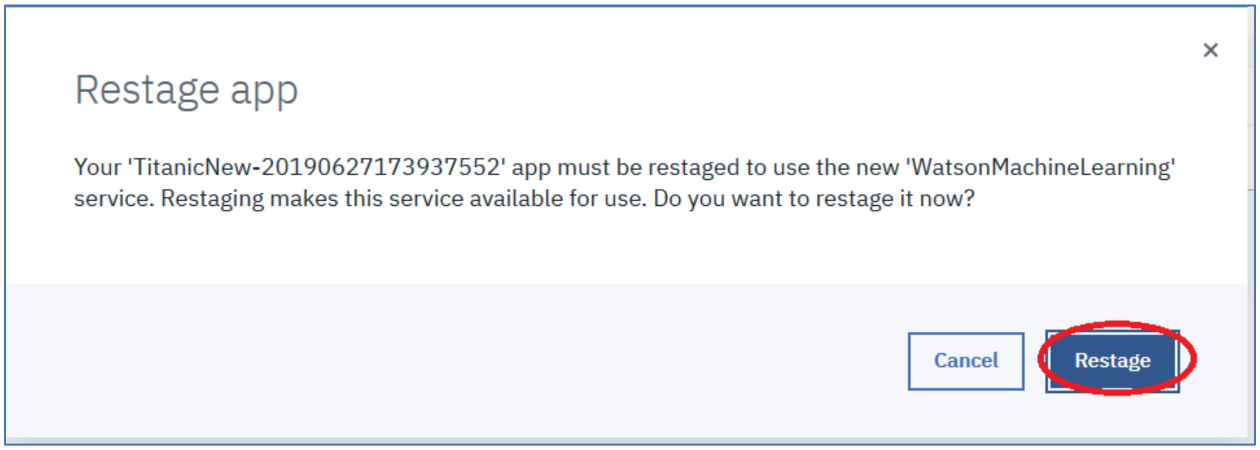
Connect IAM-Enabled Service

To connect, you can customize the ServiceID and access role used for this binding. Restaging your app is required to connect this service and may result in application downtime.

Access Role for Connection: Writer

Service ID for Connection (Optional): Auto Generate

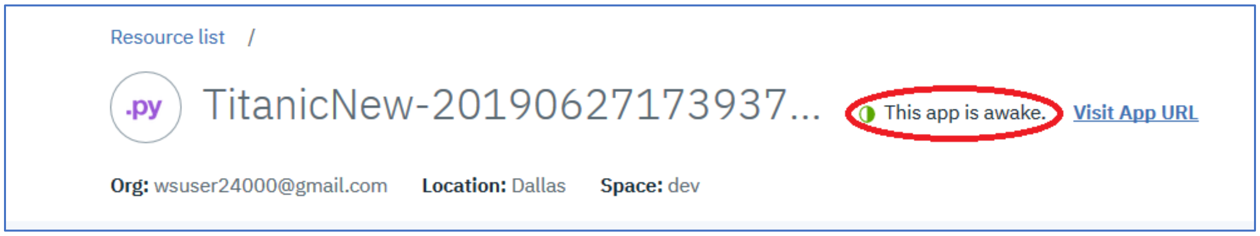
13. A **Restage app** pop up will appear. Click on **Restage**.



Restage app

Your 'TitanicNew-20190627173937552' app must be restaged to use the new 'WatsonMachineLearning' service. Restaging makes this service available for use. Do you want to restage it now?

14. Wait for the application status to change to This app is awake , or something similar.



Resource list /

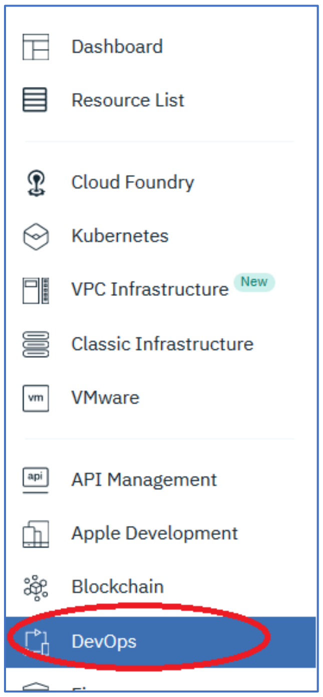
TitanicNew-20190627173937...

This app is awake.

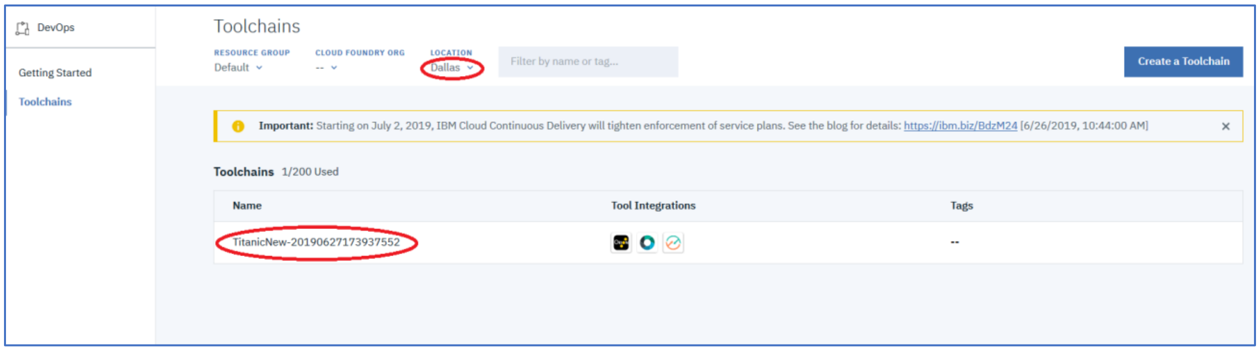
Visit App URL

Org: wsuser24000@gmail.com **Location:** Dallas **Space:** dev

15. We now have tied the web application to the Watson Machine Learning service. Note that the Watson Machine Learning service could have more than one deployed model available to select and then embed in the web application. Click on the icon and click on DevOps in the pulldown to navigate to the Toolchain.



16. We are now going to paste the scoring endpoint into the application code. Click on the Toolchain (Titanic-2019xxxxx below). If no toolchain appears, switch the region to Dallas.

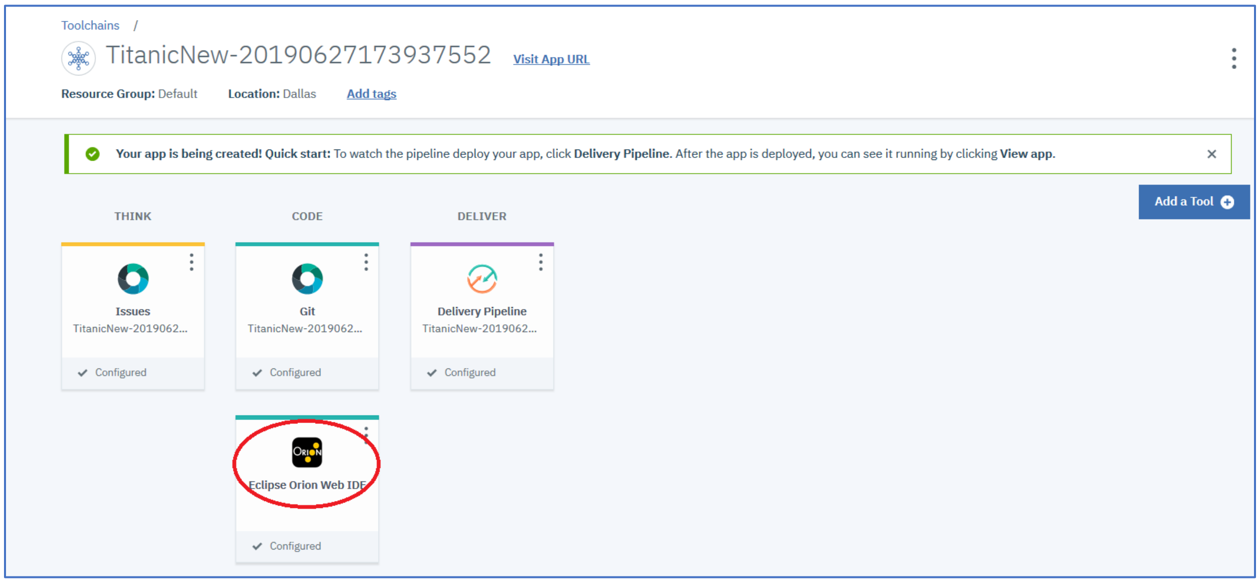


Important: Starting on July 2, 2019, IBM Cloud Continuous Delivery will tighten enforcement of service plans. See the blog for details: https://ibm.biz/BdzM24 [6/26/2019, 10:44:00 AM]

Toolchains 1/200 Used

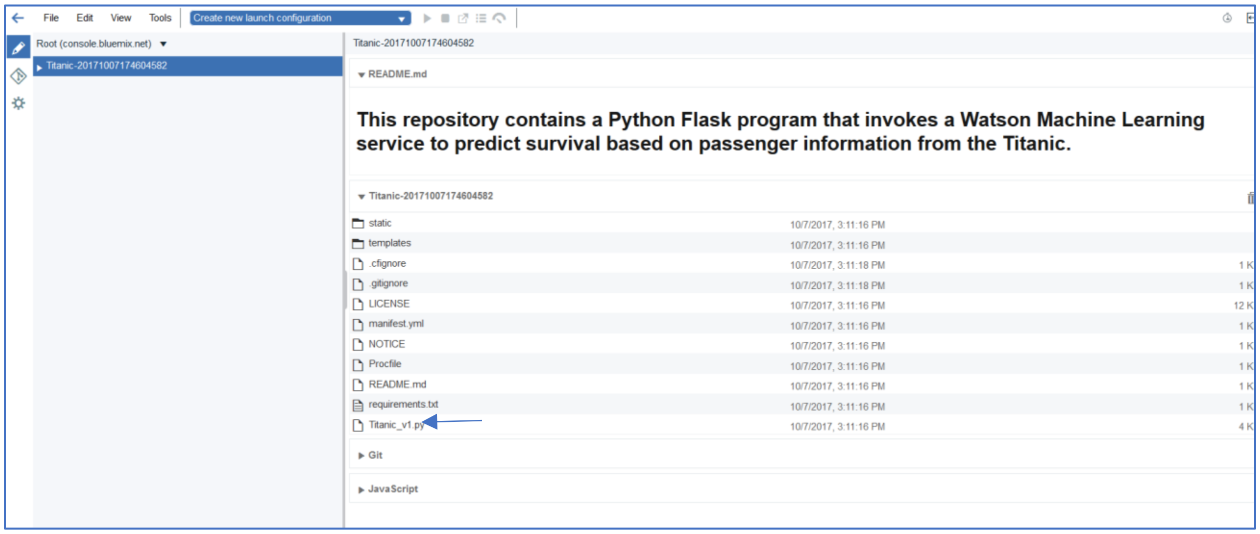
Name	Tool Integrations	Tags
TitanicNew-20190627173937552		--

17. Click on the Eclipse Orion Web IDE.



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.**

18. Click on the Titanic_v1.py file. This is a python source file.


File
Edit
View
Tools
Create new launch configuration
Root (console.bluemix.net)
Titanic-20171007174604582

Titanic-20171007174604582

README.md

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

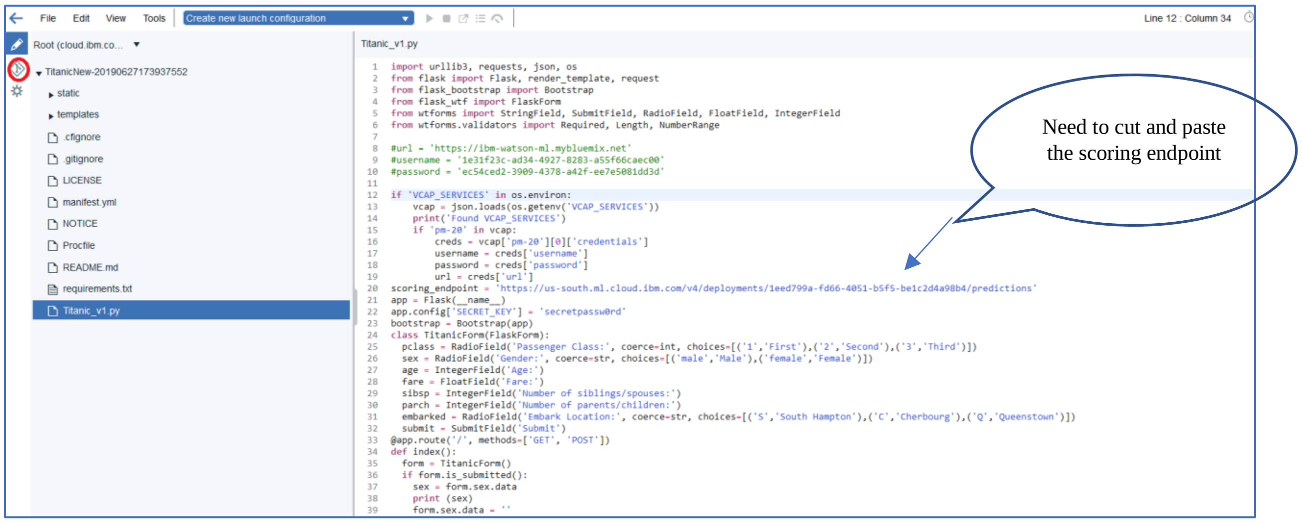
Titanic-20171007174604582

static	10/7/2017, 3:11:16 PM	
templates	10/7/2017, 3:11:16 PM	
.cfignore	10/7/2017, 3:11:18 PM	1 K
.gitignore	10/7/2017, 3:11:18 PM	1 K
LICENSE	10/7/2017, 3:11:16 PM	12 K
manifest.yml	10/7/2017, 3:11:16 PM	1 K
NOTICE	10/7/2017, 3:11:16 PM	1 K
Procfile	10/7/2017, 3:11:16 PM	1 K
README.md	10/7/2017, 3:11:16 PM	1 K
requirements.txt	10/7/2017, 3:11:16 PM	1 K
Titanic_v1.py	10/7/2017, 3:11:16 PM	4 K

Git

JavaScript

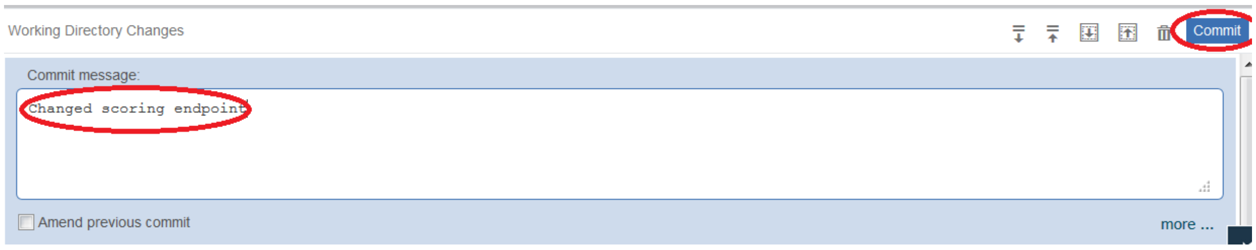
19. Look around line 20 in the Titanic_v1.py file for the "scoring endpoint =". Select the scoring endpoint value in line 20 (starting with https:// you may want to use Shift-End to get to the end of the line, and then back up one space to not select the endpoint quote – if you do just make sure to put it back in). Copy the scoring endpoint from the Watson Studio Implementation panel and use Ctrl-V to paste the scoring endpoint into the Titanic_v1.py file. Enter Ctrl-S or File > Save to save the file. Then click on the icon on the top left.



Titanic_v1.py

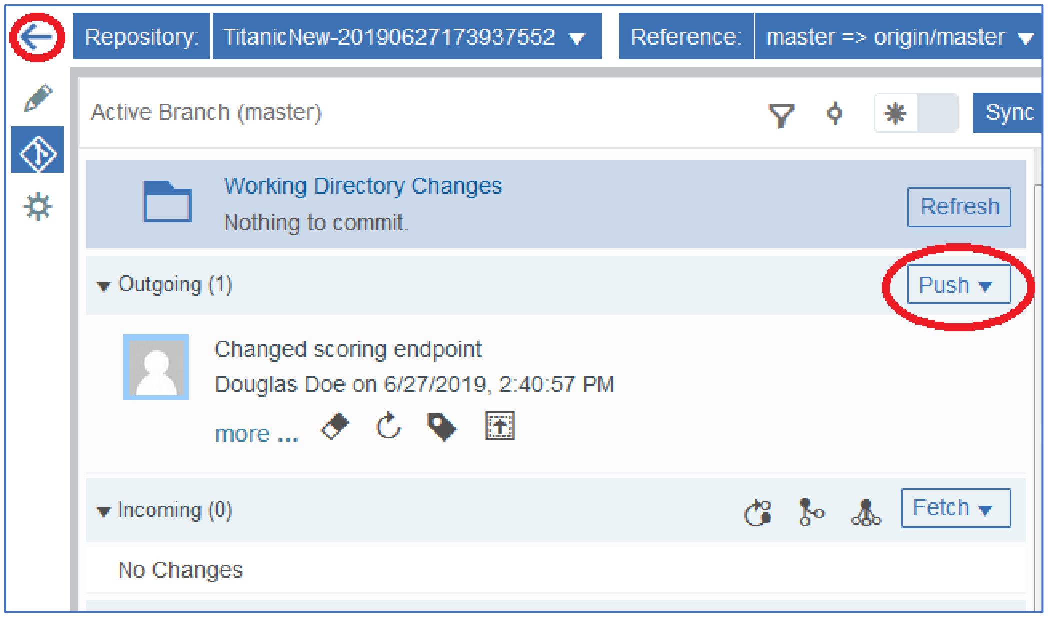
```
 1 import urllib3, requests, json, os
 2 from flask import Flask, render_template, request
 3 from flask_bootstrap import Bootstrap
 4 from flask_wtf import FlaskForm
 5 from wtforms import StringField, SubmitField, RadioField, FloatField, IntegerField
 6 from wtforms.validators import Required, Length, NumberRange
 7
 8 #url = 'https://ibm-watson-ml.mybluemix.net'
 9 #username = '1e31f23c-ad34-4927-8283-a55f66caec00'
10 #password = 'ec54ced2-3909-4378-a42f-ee7e5081dd3d'
11
12 if 'VCAP_SERVICES' in os.environ:
13     vcap = json.loads(os.getenv('VCAP_SERVICES'))
14     print('Found VCAP_SERVICES')
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 = 'https://us-south.ml.cloud.ibm.com/v4/deployments/1eed799a-fd66-4051-b5f5-be1c2d4a98b4/predictions'
21 app = Flask(__name__)
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   sex = RadioField('Gender:', coerce=str, choices=[('male','Male'),('female','Female')])
27   age = IntegerField('Age:')
28   fare = FloatField('Fare:')
29   sibsp = IntegerField('Number of siblings/spouses:')
30   parch = IntegerField('Number of parents/children:')
31   embarked = RadioField('Embark Location:', coerce=str, choices=[('S','South Hampton'),('C','Cherbourg'),('Q','Queenstown')])
32   submit = SubmitField('Submit')
33 @app.route('/', methods=['GET', 'POST'])
34 def index():
35   form = TitanicForm()
36   if form.is_submitted():
37     sex = form.sex.data
38     print (sex)
39     form.sex.data = ''
```

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



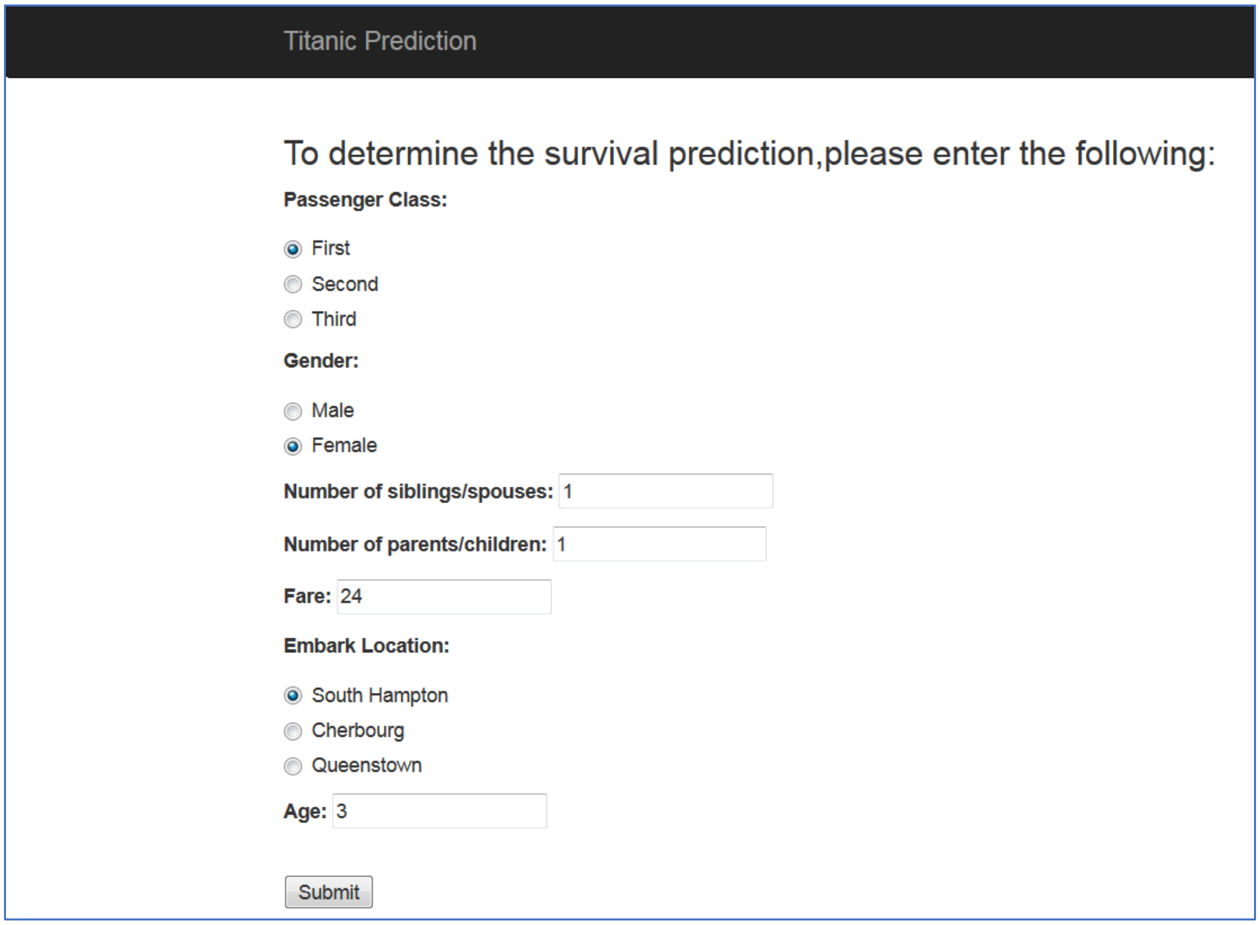
```
Changed scoring endpoint
```

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


Repository: TitanicNew-20190627173937552
Reference: master => origin/master
Active Branch (master)
Sync
Working Directory Changes
Nothing to commit.
Refresh
Outgoing (1)
Push
Changed scoring endpoint
Douglas Doe on 6/27/2019, 2:40:57 PM
more ...
Incoming (0)
Fetch
No Changes

22. Click on the **Delivery Pipeline** to view status of the deployment as before. Once the Deploy Stage status shows **Deploy passed now (**it shouldn't take longer than 2 minutes so reload the browser in case the UI didn't update), click on**View Console**, and then **Visit App URL** (refer to steps 6,7,8 above) to view the running application.

23. The web form should appear. Enter data in all the fields and click on the **Submit** button.


Titanic Prediction

To determine the survival prediction,please enter the following:

Passenger Class:

[✓ First]
Second
Third

Gender:

Male
[✓ Female]

Number of siblings/spouses: 1

Number of parents/children: 1

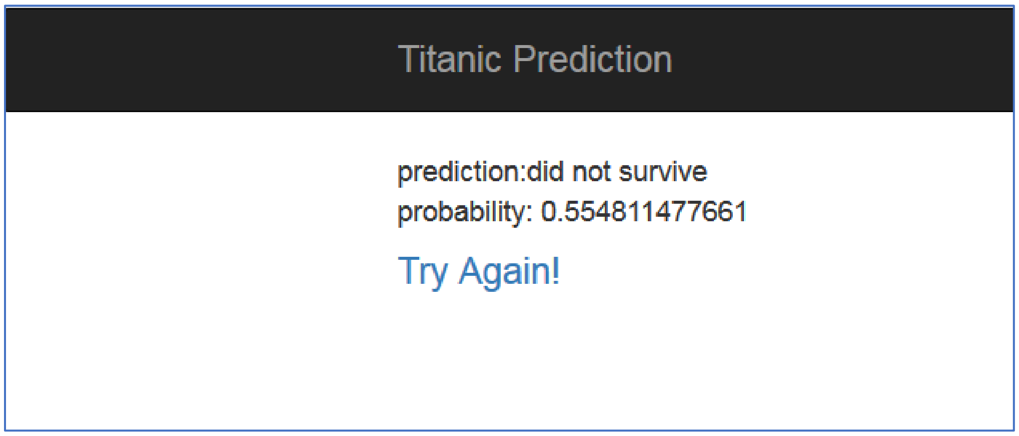
Fare: 24

Embark Location:

[✓ South Hampton]
Cherbourg
Queenstown

Age: 3

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:did not survive
probability: 0.554811477661