

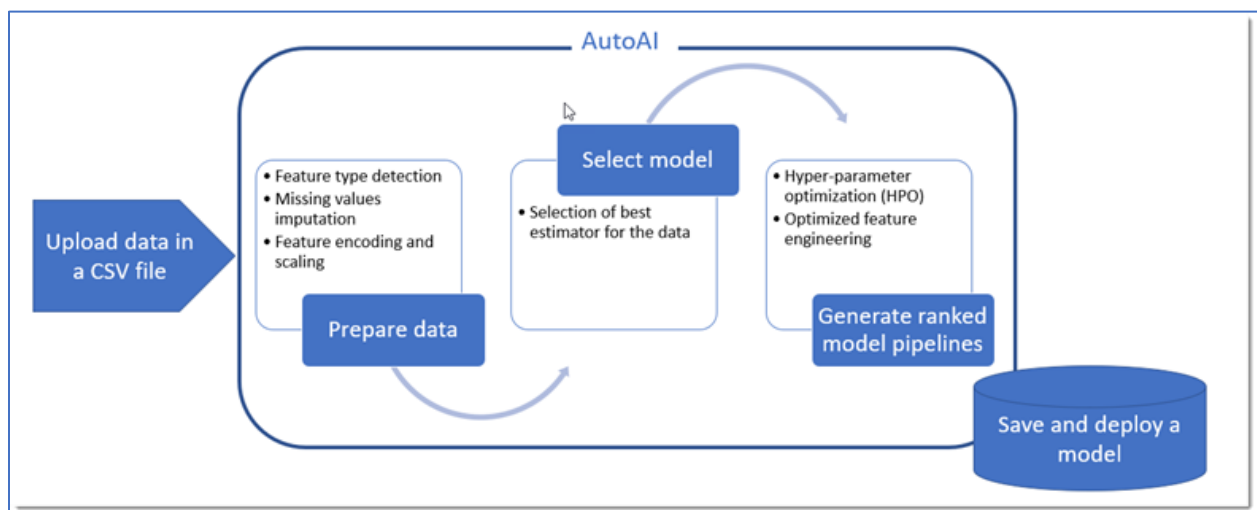
# AutoAI + DevOps

## Introduction

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 Female Human Trafficking datasets. The second part will deploy an application using the IBM Cloud DevOps toolchain that will invoke the deployed model to predict the trafficking risk.

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.



The AutoAI process follows this sequence to build candidate pipelines:

- [Data pre-processing](#)
- [Automated model selection](#)
- [Automated feature engineering](#)
- [Hyperparameter optimization](#)

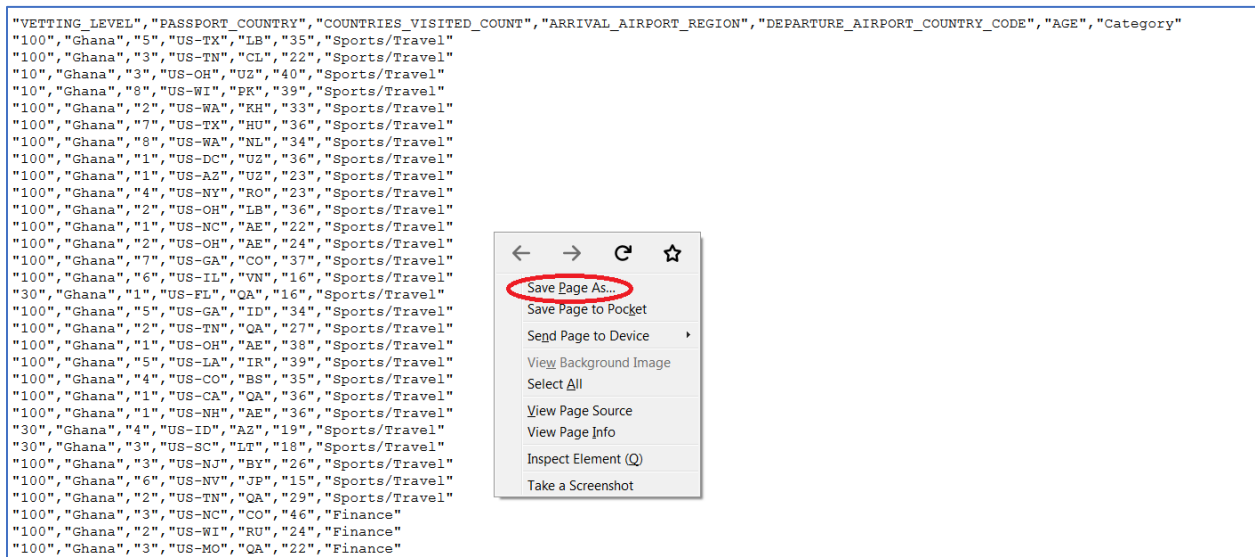
We will perform the following steps in this lab:

1. Download a female human trafficking cleansed dataset from the github repo
2. Add an Auto AI Experiment to create a model to predict the trafficking risk
3. Save and Deploy the model

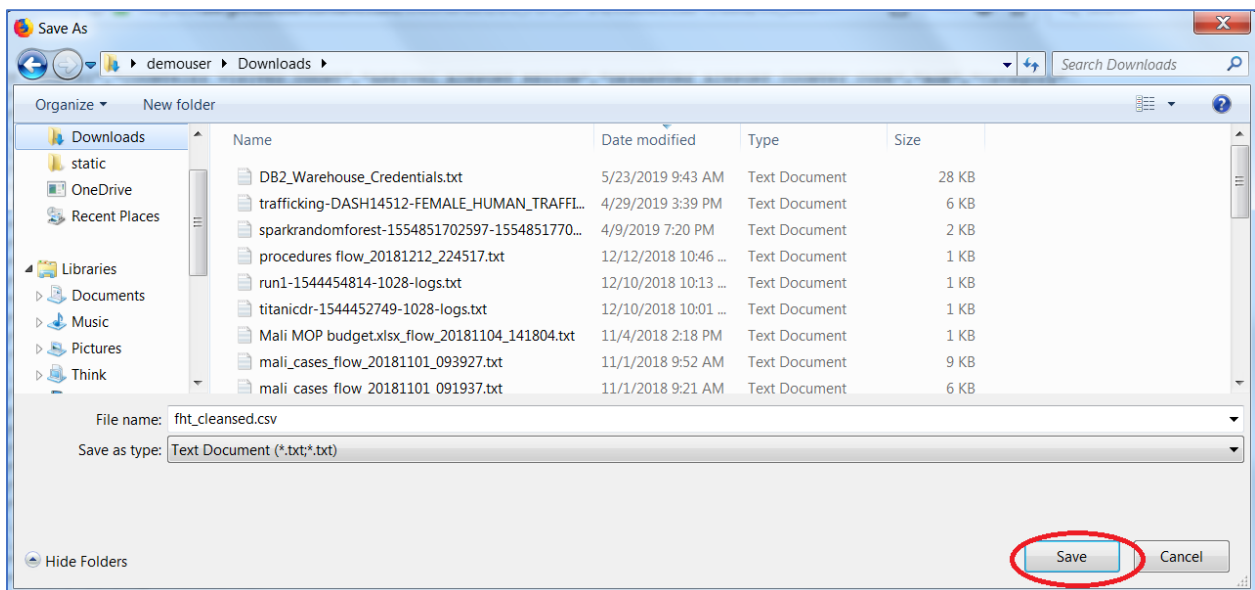
4. Test the model
5. Deploy a simple web front-end application and connecting it to the deployed model using an IBM Cloud DevOps toolchain.


## Step 1: Download a female human trafficking cleansed dataset

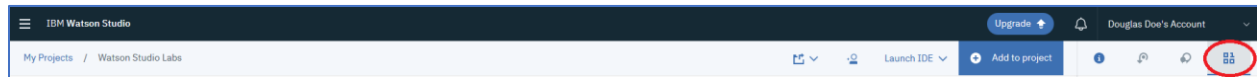
1. Download the fht\_cleansed.csv data file from the following location by clicking on the link [FHT Data](#)
2. Right-click on the window, and click **Save Page As...**



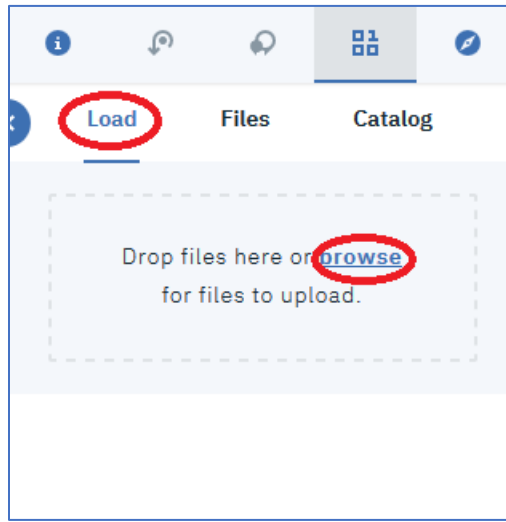
3. Click on **Save**.



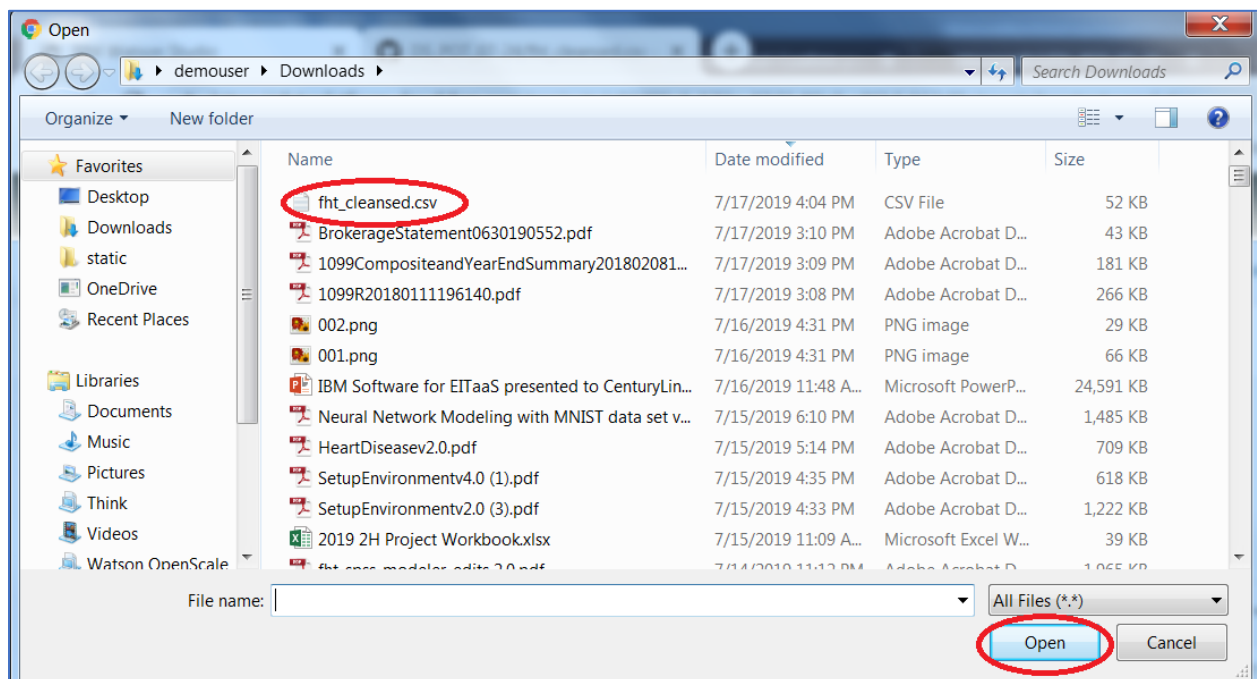
4. Go back to your Watson Studio Labs project. Click on the  icon.



5. Click on the **Load** tab and then click on **browse**. If you don't see the **Load** tab, click on the  icon again.



6. Go to the folder where the titanic\_cleansed.csv file is stored. Select the fht\_cleansed.csv file and then click **Open**.



7. The fht\_cleansed.csv file is now added as a Data Asset.

The screenshot shows the 'Assets' tab in the IBM Watson Studio interface. The breadcrumb navigation at the top reads 'My Projects / Watson Studio Labs'. Below the navigation bar, there are tabs for 'Overview', 'Assets', 'Environments', 'Jobs', 'Bookmarks', 'Deployments', 'Access Control', and 'Settings'. The 'Assets' tab is active. A search bar asks 'What assets are you looking for?'. Below it, a section titled 'Data assets' shows '0 asset selected.'. A table lists the assets with columns: NAME, TYPE, CREATED BY, LAST MODIFIED, and ACTIONS. One asset is listed: 'fht\_cleansed.csv' (circled in red), which is a 'Data Asset' created by 'Robert Doe' on '17 Jul 2019, 4:07:36 pm'.

NAME	TYPE	CREATED BY	LAST MODIFIED	ACTIONS
fht_cleansed.csv	Data Asset	Robert Doe	17 Jul 2019, 4:07:36 pm	

## Step 2: Add an AutoAI Experiment.

1. If not on the **Assets** page, click on the **Assets** Tab

The screenshot shows the IBM Watson Studio interface with the 'Assets' tab selected and circled in red. The breadcrumb navigation at the top reads 'My Projects / Watson Studio Labs'. The navigation bar includes 'Overview', 'Assets', 'Environments', 'Bookmarks', 'Deployments', 'Access Control', and 'Settings'.

2. Click on **Add to project**.

The screenshot shows the IBM Watson Studio interface with the 'Add to project' button circled in red. The breadcrumb navigation at the top reads 'My Projects / Watson Studio Labs'. The navigation bar includes 'Overview', 'Assets', 'Environments', 'Bookmarks', 'Deployments', 'Access Control', and 'Settings'.

3. Click on **AutoAI Experiment**.

The screenshot shows the 'Choose asset type' dialog box. It lists various available asset types in a grid. The 'AutoAI experiment' option is circled in red. The dialog box has a 'Close' button at the bottom right.

Choose asset type

AVAILABLE ASSET TYPES

- Data
- Connection
- Connected data
- AutoAI experiment
- Notebook
- Dashboard
- Visual Recognition ...
- Natural Language CL...
- Watson Machine Lea...
- Experiment
- Modeler flow
- Data Refinery flow
- Streams flow
- Synthesized neural n...

Close

4. 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 \*

FHT\_AutoAI

Description

Description of AutoAI experiment

#### Associated services

Watson Machine Learning Service Instance \*

WatsonMachineLearning

Compute configuration \* [i](#)

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.


Cancel Create

5. Click on **Select from project**.

My Projects / Watson Studio Labs / FHT\_AutoAI

CONFIGURE AUTOAI EXPERIMENT  
FHT\_AutoAI

#### Add training data



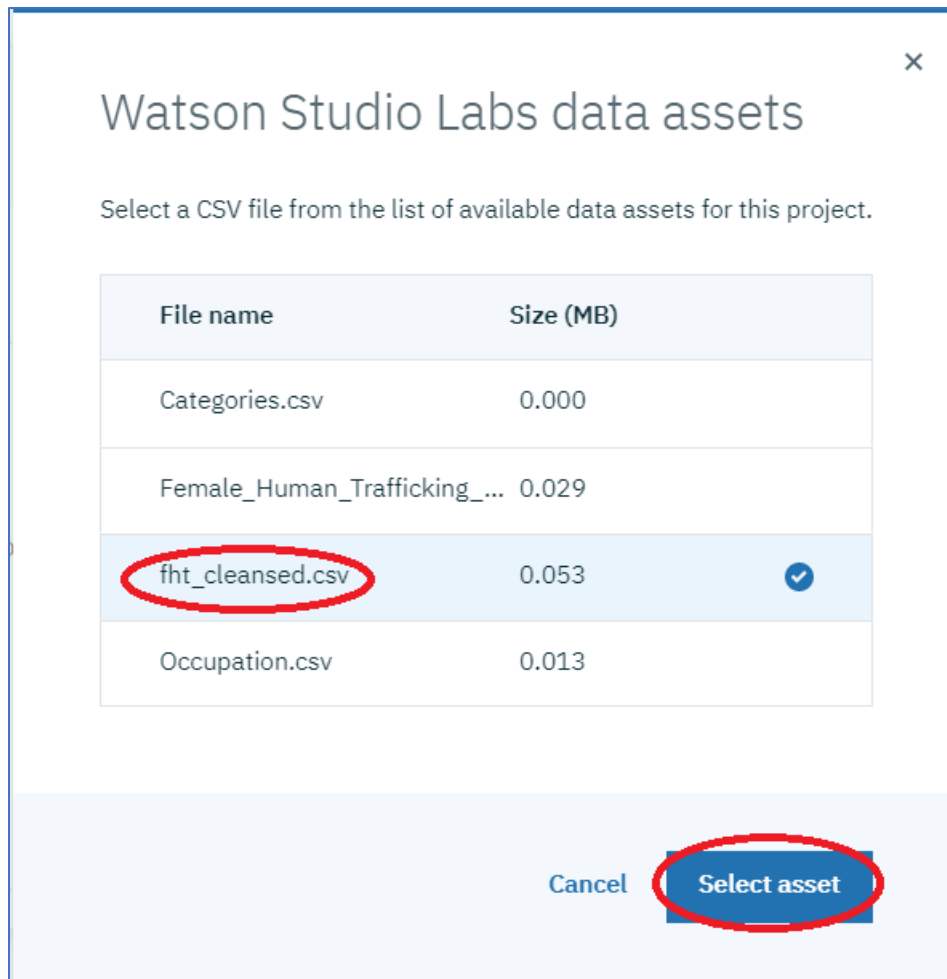
Drop a .csv file here or [browse](#) for a file to upload.

Select from project

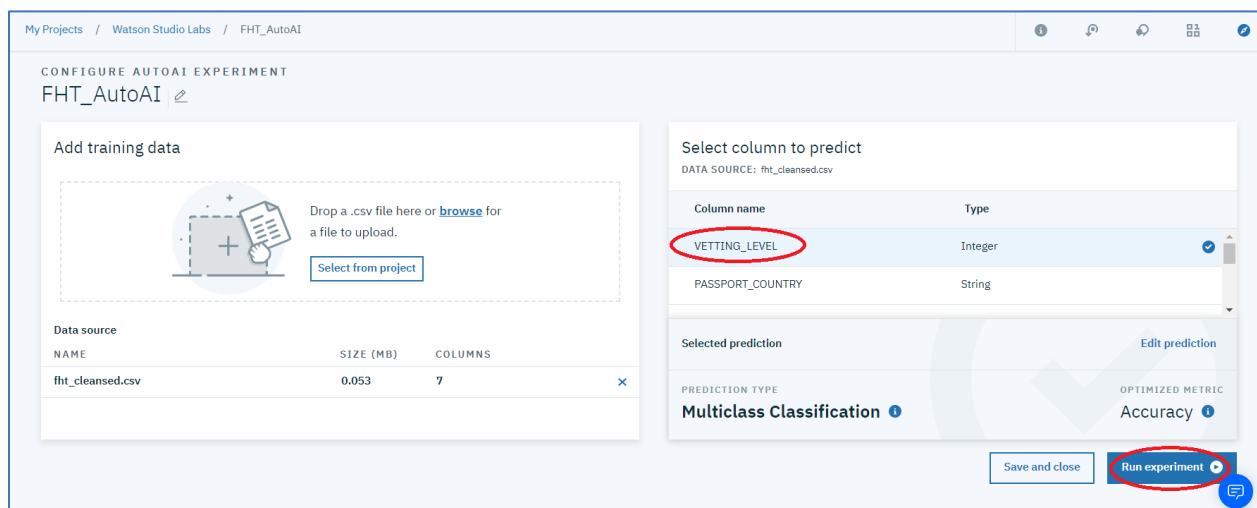
Save and close


Run experiment

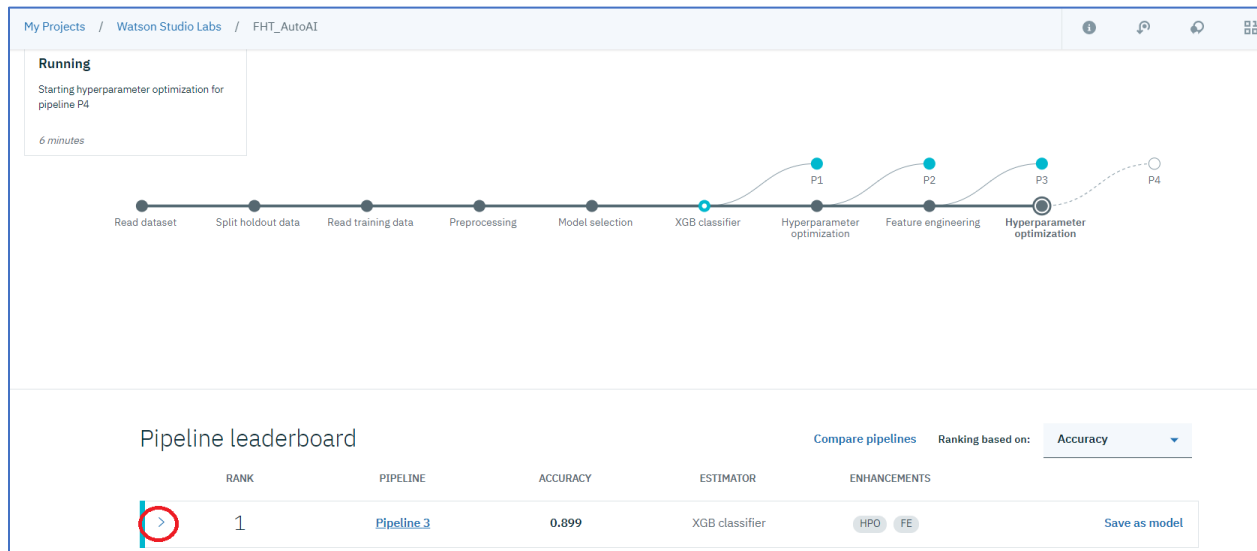
6. Select the **fht\_cleansed.csv** dataset and click on **Select asset**.



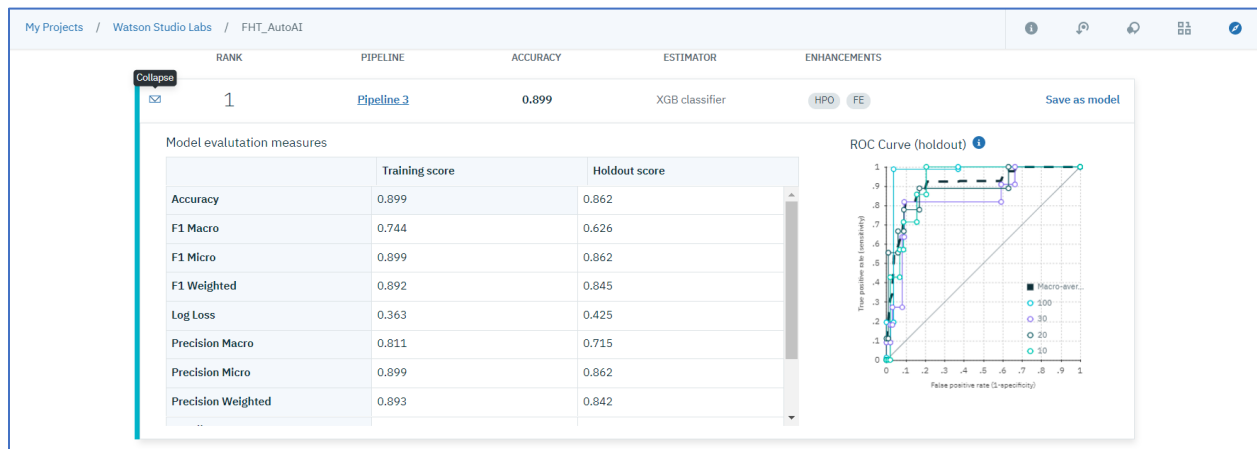
7. Select **VETTING\_LEVEL** as the column to predict, leave the default for **OPTIMIZED METRIC** and click on **Run Experiment**. Note the system scanned the **VETTING\_LEVEL** values to determine that a **Multiclass Classification** was the **PREDICTION TYPE**.



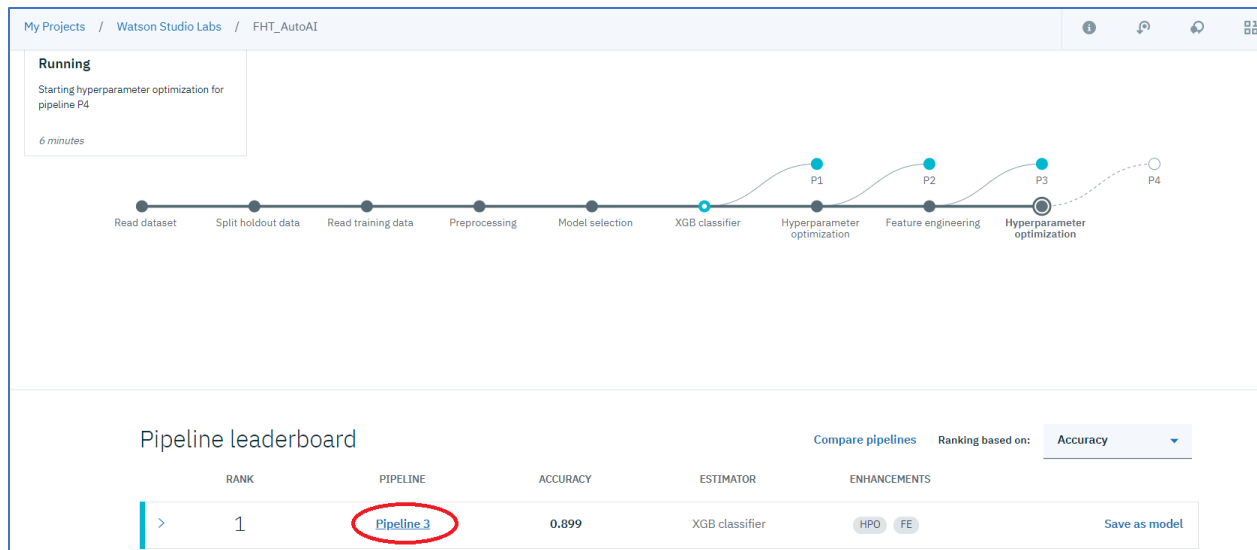
8. Please wait several minutes (until at Pipeline 3 has been created) as the alternative solutions are analyzed. The pipeline summary is then displayed. Click on the right arrow  next to the first pipeline (Pipeline 3).



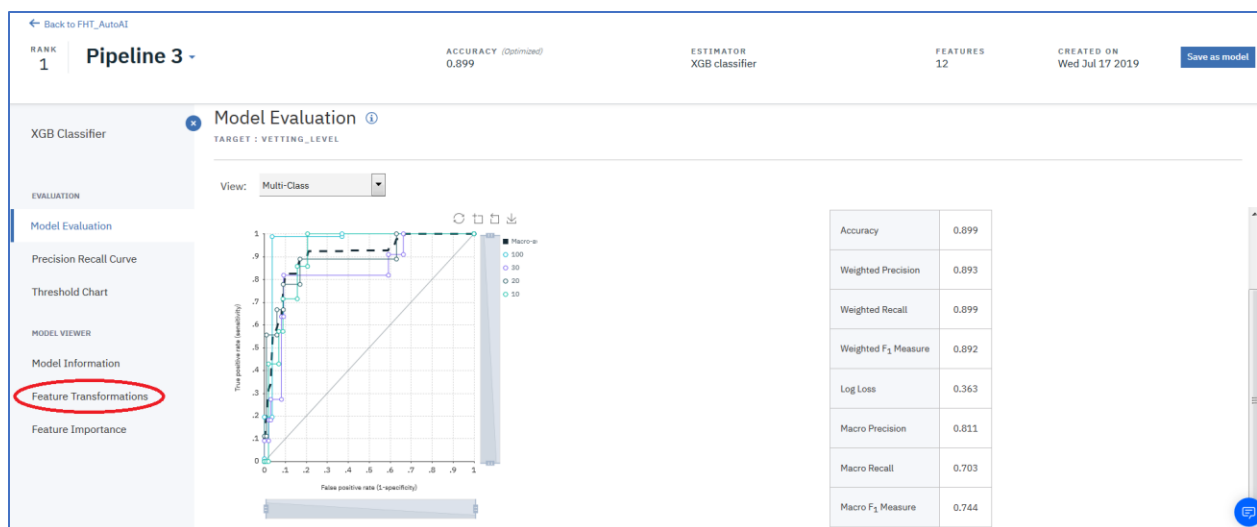
9. Scores are displayed for different metrics for both the training sample and the holdout sample.



10. Click on **Pipeline 3**



11. The model evaluation metrics for the training sample are displayed. On the left are options for additional information. Click on **Feature Transformations**.



12. We see there are now 12 features. The 6 original features, plus 6 transformed features. The transformed features are used to improve the classification metrics. Click on **Feature Importance**.



← Back to FHT\_AutoAI

RANK 1 Pipeline 3 ACCURACY (Optimized) 0.899 ESTIMATOR XGB classifier FEATURES 12 CREATED ON Wed Jul 17 2019

XGB Classifier

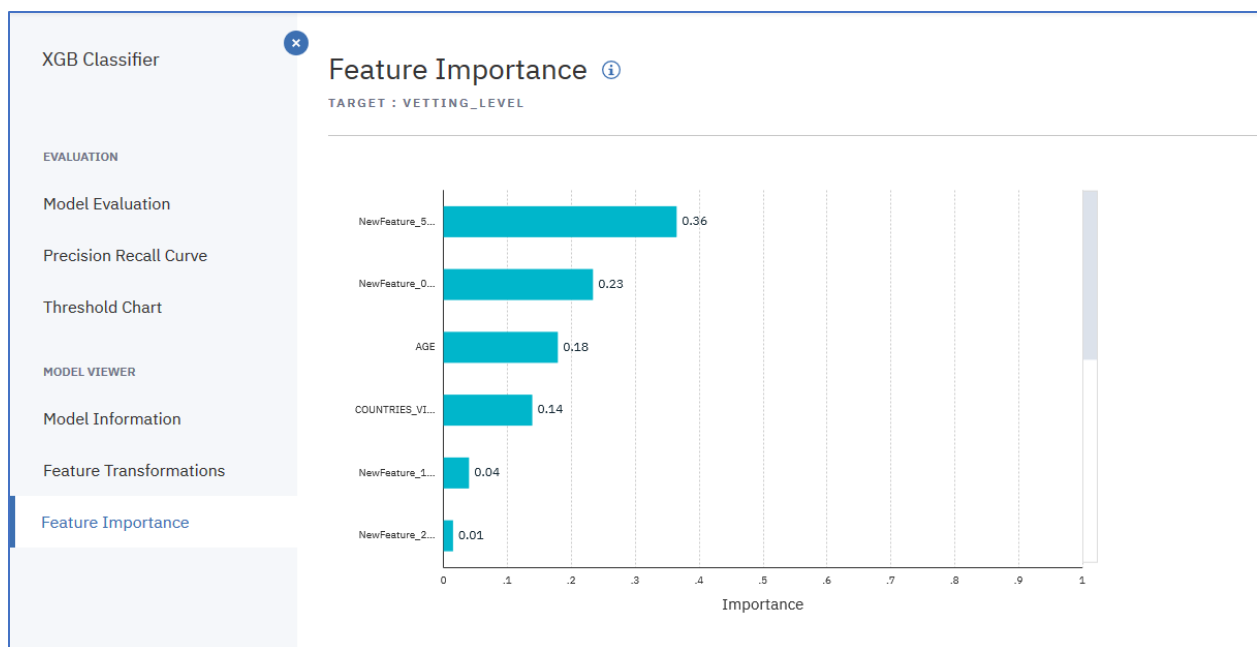
Feature Transformations ⓘ

TARGET : VETTING\_LEVEL

New Feature	Original Feature	Transformation
NewFeature_0	PASSPORT_COUNTRY	sine(PASSPORT_COUNTRY)
NewFeature_1	COUNTRIES_VISITED_COUNT	sine(COUNTRIES_VISITED_COUNT)
NewFeature_2	ARRIVAL_AIRPORT_REGION	sine(ARRIVAL_AIRPORT_REGION)
NewFeature_3	DEPARTURE_AIRPORT_COUNTRY_CODE	sine(DEPARTURE_AIRPORT_COUNTRY_CODE)
NewFeature_4	AGE	sine(AGE)
NewFeature_5	Category	sine(Category)

Feature Importance

13. According to **Feature Importance**, the transformed Category feature is the most important, followed by the transformed Passport Country feature.



## Step 3: Save and Deploy the Model

1. Click on **Save as model**

← Back to FHT\_AutoAI

RANK 1 Pipeline 3 ACCURACY (Optimized) 0.899 ESTIMATOR XGB classifier FEATURES 12 CREATED ON Wed Jul 17 2019

Save as model

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

×

## Save as model

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

**Model name**

FHT\_AutoAI - P3 XGBClassifierEstimator

**Description (optional)**

Description of model

Cancel

Save

- A message is displayed showing the model was successfully saved. Click on **View in project**.

IBM Watson Studio

Upgrade

Robert Doe's Account

My Projects / Watson Studio Labs / FHT\_AutoAI

← Back to FHT\_AutoAI

RANK 1	Pipeline 3	ACCURACY (Optimized) 0.899	ESTIMATOR XGB classifier	FEATURES 12	CREAT Wed 3
-----------	------------	-------------------------------	-----------------------------	----------------	----------------

XGB Classifier

Feature Importance

'FHT\_AutoAI - P3 XGBClassifierEstimator' was successfully saved to 'Watson Studio Labs'.

view in project

- Click on **Deployments**.

MODEL

FHT\_AutoAI - P3 XGBClassifierEstimator

Overview

Evaluation

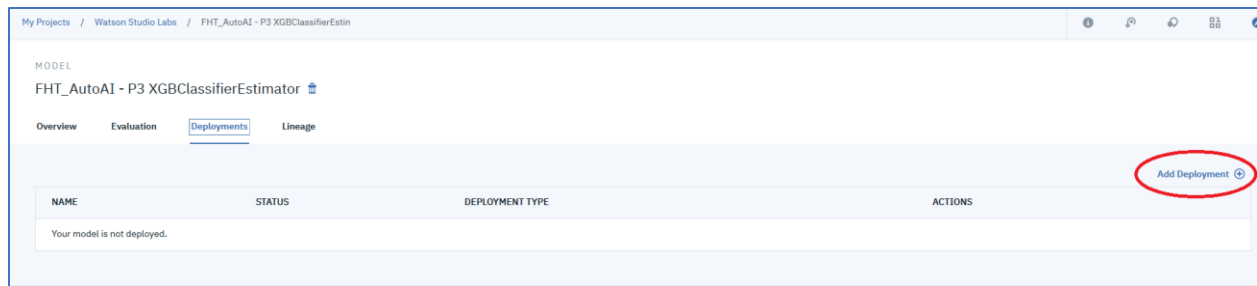
Deployments

Lineage

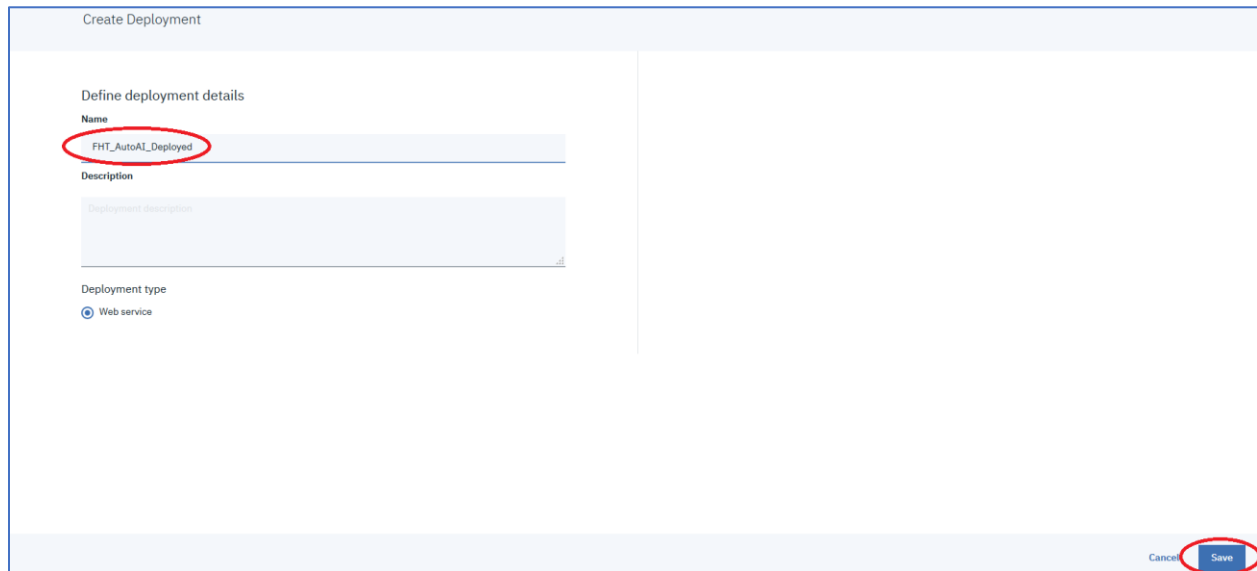
Add Deployment

NAME	STATUS	DEPLOYMENT TYPE	ACTIONS
Your model is not deployed.			

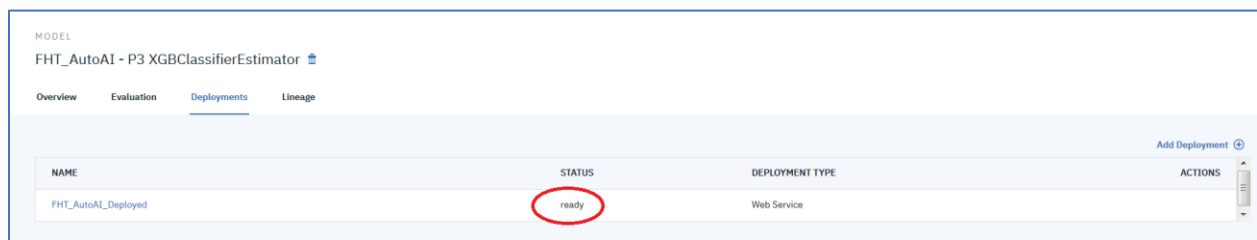
- Click on **Add Deployment**



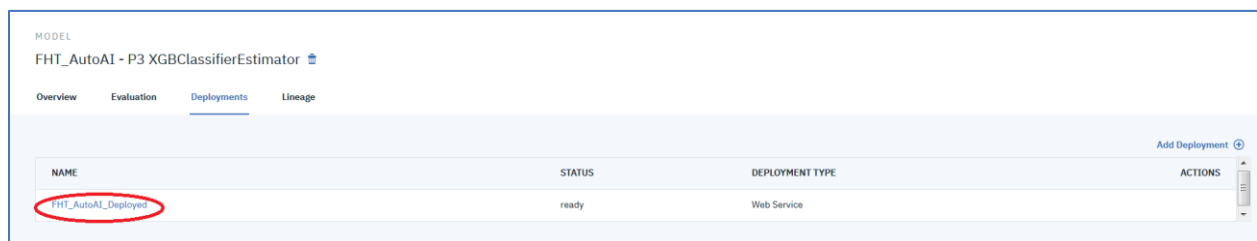
6. Enter a **Name**, optionally a **Description** and click **Save**.



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



8. Click on **FHT\_AutoAI\_Deployed**.



9. Click on **Implementation**.

FHT_AutoAI_Deployed	
<a href="#">Overview</a> <a href="#">Implementation</a> <a href="#">Test</a>	
Deployment	
Name	FHT_AutoAI_Deployed
Type	Web Service
Deployment ID	e020c2c0-7c2a-43fe-9204-58f6398990bb
Status	ready
Asset type	model
Asset name	FHT_AutoAI - P3 XGBClassifierEstimator
Machine learning service	WatsonMachineLearning
Created	17 Jul 2019 10:04pm
Last modified	17 Jul 2019 10:08pm

10. The Implementation panel provides information for the application developers to invoke the deployed model. It includes sample code in various programming languages and the scoring endpoint to be used when invoking the web service. Open Windows Notepad to copy and paste the scoring endpoint, or just leave this panel available to cut and paste the scoring endpoint. We will need the scoring endpoint in the next section.

FHT_AutoAI_Deployed	
<a href="#">Overview</a> <a href="#">Implementation</a> <a href="#">Test</a>	
Implementation <span>View API Specification</span>	
Scoring End-point	<a href="https://us-south.ml.cloud.ibm.com/v4/deployments/e020c2c0-7c2a-43fe-9204-58f6398990bb/predictions">https://us-south.ml.cloud.ibm.com/v4/deployments/e020c2c0-7c2a-43fe-9204-58f6398990bb/predictions</a>
Authorization: Bearer <token>	Review the <a href="#">WML authentication</a> documentation for details about generating IAM tokens.
ML-Instance-ID	The "ML-Instance-ID" HTTP header must be populated with the WML instance id, which can be obtained as <a href="#">described here</a> .
Content-type: application/json	Required if the request body is sent in JSON format.
Code Snippets	
cURL	<pre>curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'Authorization: Bearer \$IAM_TOKEN' --header 'ML-Instance-ID: \$ML_INSTANCE_ID' -d '{"input_data": [{"fields": [FARRAY_OF_FEP</pre>

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

This section provides an example of a simple Python Flask web front-end application that invokes the Female Human Trafficking risk prediction deployed model, 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.

The steps below guide you in configuring the application to connect to your Watson Machine Learning service, and to update the application with the deployed model's 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. 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:** ⓘ  
FHT-20190516021126700

**IBM Cloud API Key:** ⓘ  
IBM Cloud API Key Create +

The value is required.

Region	Organization	Space
The IBM Cloud CF region	The IBM Cloud CF org	The IBM Cloud CF space
The value is required.	The value is required.	The value is required.

3. 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 FHT-20190516021126700

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

Cancel Create

4. Please wait until the Region, Organization, and Space are filled in. **Note that these must match the corresponding Region, Organization, and Space where the FHT\_WML\_Model\_Deployed is deployed.** Switch the **Region** to Dallas (Production), 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:**

**IBM Cloud API Key:**  [Create +](#)


**Region:**

**Organization:**


**Space:**

[Deploy](#)


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

 FHT-20190516022609200


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 created, click **View Console**.


THINK


  
**Issues**  
FHT-2019051602260...  
✓ Configured

CODE

  
**Git**  
FHT-2019051602260...  
✓ Configured

DELIVER

  
**Delivery Pipeline**  
FHT-2019051602260...  
✓ Configured

  
**Eclipse Orion Web IDE**  
✓ Configured

6. 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**

The screenshot shows two side-by-side panels for a CI/CD pipeline. The left panel is titled 'Build Stage' and shows a 'STAGE PASSED' status. It includes a 'LAST INPUT' section with a commit by 'bleonardb3' and a 'JOBS' section showing a 'Build' job that passed 3 minutes ago. The right panel is titled 'Deploy Stage' and also shows a 'STAGE PASSED' status. It includes a 'LAST INPUT' section showing 'Build 2' and a 'JOBS' section showing a 'Deploy' job that passed 'now'. A red circle highlights the 'Deploy Passed now' status. Below the jobs, the 'LAST EXECUTION RESULT' section shows a build ID 'FHT-20190516022609200' with a 'view console' link circled in red.

7. Click on **Visit App URL**

The screenshot shows a resource page for an application. At the top, it says 'Resource list /'. Below this, there is a card for the application 'FHT-20190516022609200'. The card includes a '.py' icon, the application name, a status indicator 'This app is awake.', and a 'Visit App URL' link circled in red. At the bottom of the card, it shows 'Org: wsuser19000@gmail.com', 'Location: Dallas', and 'Space: dev'.

8. The web form collecting the FHT 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 FHT Prediction browser tab.



To determine the trafficking risk prediction, please enter the following:

Categories

Sports/Travel

Age:

Number of countries visited:

Passport Country

- ☐ Ghana
- ☐ Brazil
- ☐ Pakistan
- ☐ Bangladesh
- ☐ Haiti
- ☐ India

Submit

9. 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**.

Connections

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

Create connection

10. You should see at least 3 services listed, a Cloud Object Storage service, a Spark service, and a Watson Machine Learning service. Point the cursor on the **Machine Learning** service for your application, and then click on **Connect**.

### Connect Existing Compatible Service

Q Search compatible services X

All Resources v

10 Items per page | 1-3 of 3 items 1 of 1 pages < 1 >

SERVICES	RESOURCE GROUP	PLAN	SERVICE OFFERING
cloud-object-storage-vg	Default	Lite	Cloud Object Storage
Machine Learning	Default	Lite	Machine Learning
Spark	Default	Lite	Analytics Engine

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

### Connect IAM-Enabled Service

X

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** i

Writer v

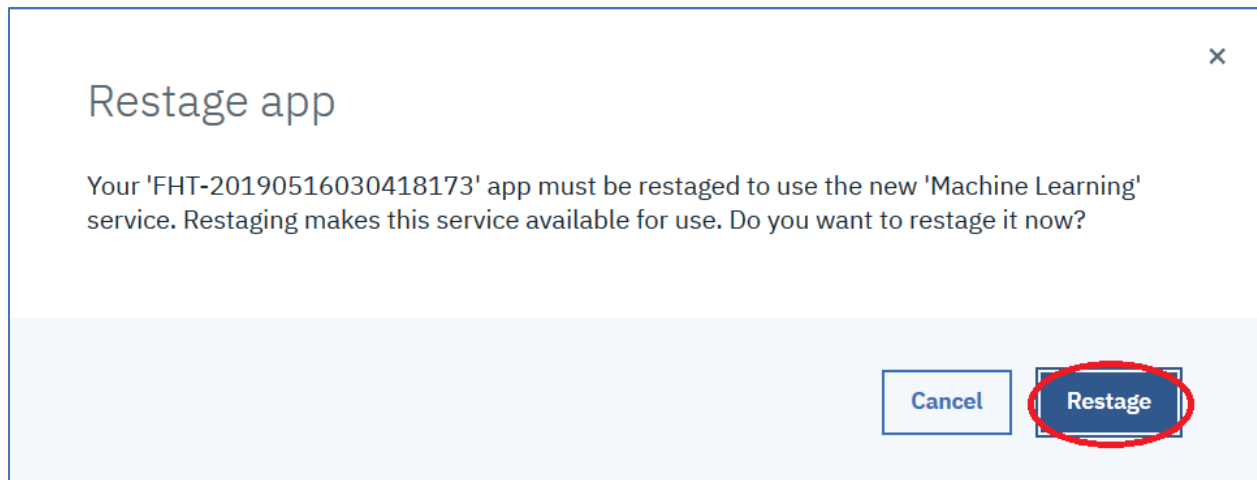
**Service ID for Connection (Optional)** i


Auto Generate v

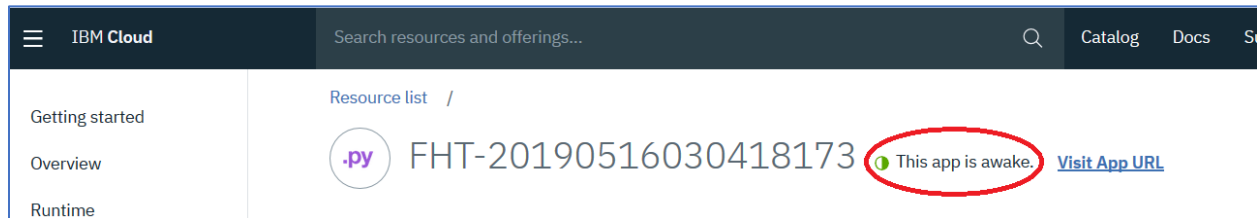
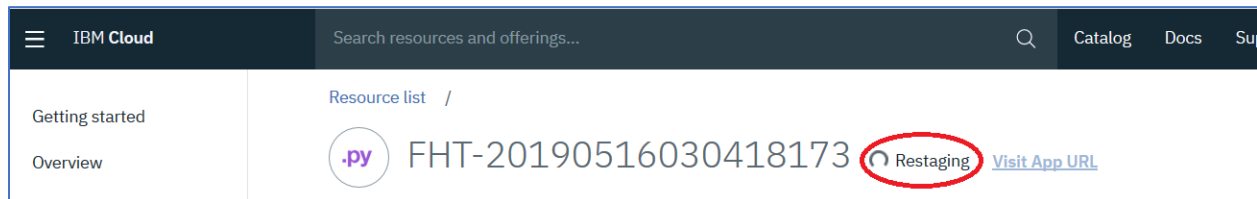
Cancel


Connect

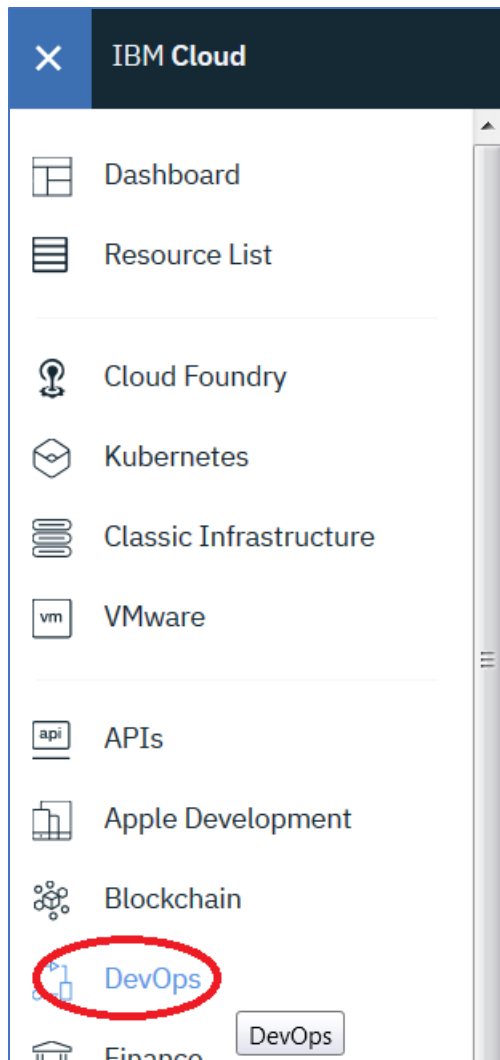
12. A **Restage app** pop up will appear. Click on **Restage**.



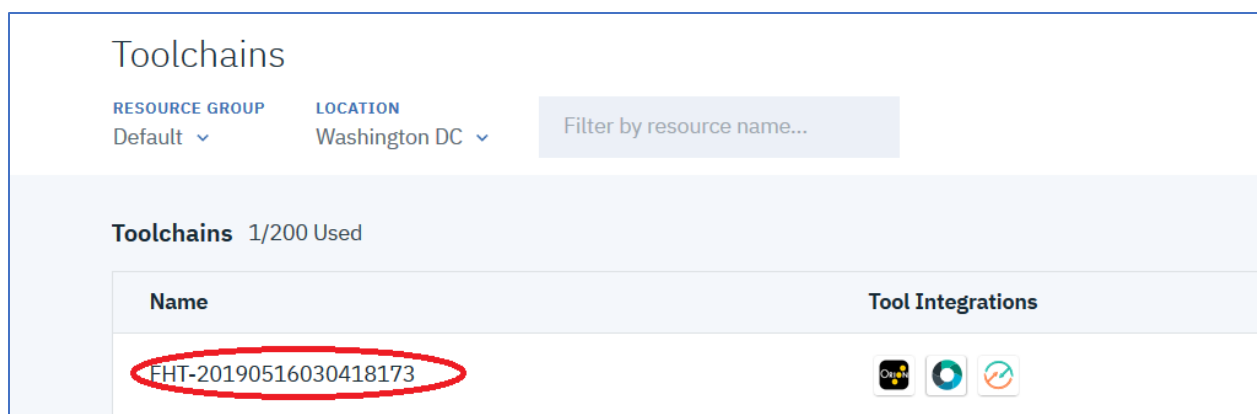
13. Wait for the application status to change from **Restaging** to  This app is awake , or something similar.



14. 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. We now need to copy the scoring endpoint, which we previously copied and pasted into Notepad, and paste it in the web application code. Click on the  icon and click on DevOps in the pulldown to navigate to the Toolchain.




15. We are now going to paste the scoring endpoint into the application code. Click on the Toolchain (FHT-2019xxxxx below).




16. Click on the Eclipse Orion Web IDE. The IDE will enable the editing of the source code to update the scoring endpoint url.


Toolchains /

 FHT-20190516030418173 [Visit App URL](#)

Resource Group: Default Location: Washington DC [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 app.

THINK




**Issues**  
FHT-2019051603041...  
✓ Configured

CODE




**Git**  
FHT-2019051603041...  
✓ Configured

DELIVER



**Delivery Pipeline**  
FHT-2019051603041...  
✓ Configured

 **Eclipse Orion Web IDE**


17. Click on the FHT.py file. This is a python source file.

FHT-20190516030418173


▼ README.md

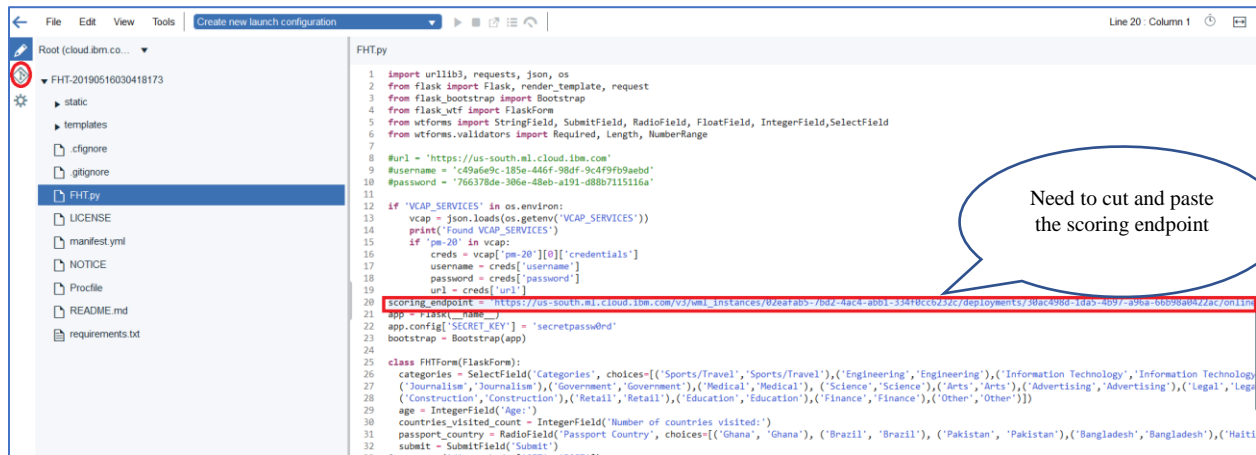
**This repository contains a Python Flask program that invokes a Watson Machine Learning service to predict trafficking risk based on travel itinerary data**

▼ FHT-20190516030418173

Name	Date Modified	Size
static	5/15/2019, 11:37:15 PM	--
templates	5/15/2019, 11:37:15 PM	--
.cignore	5/15/2019, 11:37:16 PM	1 KB
.gitignore	5/15/2019, 11:37:16 PM	1 KB
 <b>FHT.py</b>	5/15/2019, 11:37:15 PM	4 KB
LICENSE	5/15/2019, 11:37:15 PM	12 KB
manifest.yml	5/15/2019, 11:37:15 PM	1 KB
NOTICE	5/15/2019, 11:37:15 PM	1 KB
Profile	5/15/2019, 11:37:15 PM	1 KB
README.md	5/15/2019, 11:37:15 PM	1 KB
requirements.txt	5/15/2019, 11:37:15 PM	1 KB

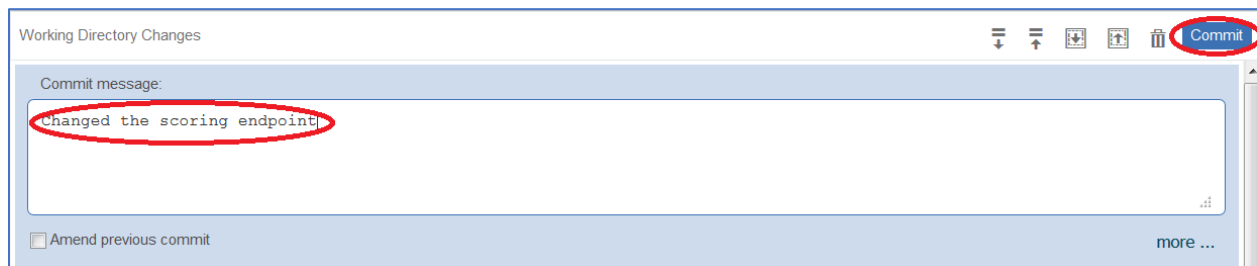
18. Go back to the Notepad file and copy the scoring endpoint to the clipboard. Look around line 20 in the FHT.py file for the “scoring endpoint =”. Select the scoring endpoint value in line 20 (starting with https:// may want to use Shift-End to get to the end of the

line, and then back up one space to not select the endpoint quote – if you do just make sure to put it back in). Enter Ctrl-V to paste the new scoring endpoint from your Notepad file. Enter Ctrl-S or File > Save to save the file. Then click on the  icon on the top left.

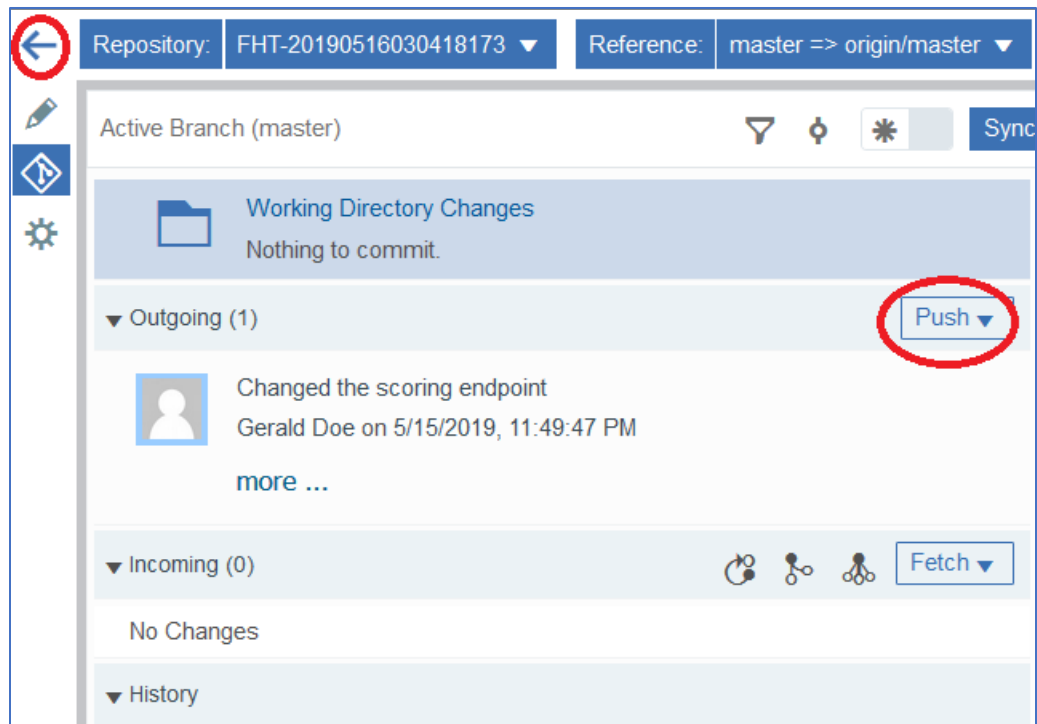


```
1 import urllib3, requests, json, os
2 from flask import Flask, render_template, request
3 from flask_bootstrap import Bootstrap
4 from flask_wtf import FlaskForm
5 from wtforms import StringField, SubmitField, RadioField, FloatField, IntegerField, SelectField
6 from wtforms.validators import Required, Length, NumberRange
7
8 #url = 'https://us-south-ml.cloud.ibm.com'
9 #username = 'c09a6e9c-185e-446f-9c4f-9b9aebd'
10 #password = '766378de-306e-48eb-a191-d88b7115116a'
11
12 if 'VCAP_SERVICES' in os.environ:
13     vcap = json.loads(os.getenv('VCAP_SERVICES'))
14     print('Found VCAP_SERVICES')
15     if 'pe-20' in vcap:
16         creds = vcap['pe-20'][0]['credentials']
17         username = creds['username']
18         password = creds['password']
19         url = creds['url']
20
21 scoring_endpoint = 'https://us-south-ml.cloud.ibm.com/v3/ml_instances/62eafab5-b02-4ac8-a081-394f8c6252c/deployments/30ac398d-1da5-4097-a96a-b0898a842zac/online'
22
23 app = Flask(__name__)
24 app.config['SECRET_KEY'] = 'secretpasswd'
25 bootstrap = Bootstrap(app)
26
27 class FHTForm(FlaskForm):
28     categories = SelectField('Categories', choices=[('Sports/Travel', 'Sports/Travel'), ('Engineering', 'Engineering'), ('Information Technology', 'Information Technology'), ('Journalism', 'Journalism'), ('Government', 'Government'), ('Medical', 'Medical'), ('Science', 'Science'), ('Arts', 'Arts'), ('Advertising', 'Advertising'), ('Legal', 'Legal'), ('Construction', 'Construction'), ('Retail', 'Retail'), ('Education', 'Education'), ('Finance', 'Finance'), ('Other', 'Other')])
29     age = IntegerField('Age:')
30     countries_visited_count = IntegerField('Number of countries visited:')
31     passport_country = RadioField('Passport Country', choices=[('Ghana', 'Ghana'), ('Brazil', 'Brazil'), ('Pakistan', 'Pakistan'), ('Bangladesh', 'Bangladesh'), ('Haiti', 'Haiti')])
32     submit = SubmitField('Submit')
```


19. 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**.




20. 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.




21. Click on the **Delivery Pipeline** to view status of the deployment as before.

 FHT-20190516030418173 [Visit App URL](#)


Resource Group: Default      Location: Washington DC      [Add tags](#)

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


THINK


  
**Issues**  
FHT-2019051603041...  
✓ Configured

CODE

  
**Git**  
FHT-2019051603041...  
✓ Configured

DELIVER

  
**Delivery Pipeline**  
FHT-2019051603041...  
✓ Configured

  
**Eclipse Orion Web IDE**  
✓ Configured

22. Once the Deployment status shows **Deploy passed now** it shouldn't take longer than 2 minutes (reload the browser in case the UI didn't update after 2 minutes). Click on **View Console**.



The screenshot displays two stages of a Jenkins pipeline: 'Build Stage' and 'Deploy Stage'. Both stages are marked as 'STAGE PASSED' with a green bar at the top. The 'Build Stage' shows a 'LAST INPUT' from 'Gerald Doe' and a 'JOBS' section with 'Build' passed 4m ago. The 'Deploy Stage' shows a 'LAST INPUT' from 'Stage: Build Stage / Job: B...' and a 'JOBS' section with 'Deploy' passed 2m ago. In the 'Deploy Stage', the 'Deploy' job is circled in red. Below the jobs, the 'LAST EXECUTION RESULT' for 'Build 2' is shown, with a 'View console' link circled in red.

23. Click on **Visit App URL**.

The screenshot shows the Jenkins application resource page. At the top, it says 'Resource list /'. Below this, there is a large ID 'FHT-20190516030418173' next to a green dot and the text 'This app is awake.' To the right of this is a 'Visit App URL' link circled in red. Below the ID, it shows 'Org: wsuser19000@gmail.com', 'Location: Dallas', and 'Space: dev'.

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

To determine the trafficking risk prediction, please enter the following:

**Categories**

Information Technology

Age: 24

Number of countries visited: 4

**Passport Country**

☒ Ghana

☐ Brazil

☐ Pakistan

☐ Bangladesh

☐ Haiti

☐ India

Submit

25. 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.

FHT Prediction

prediction: low risk  
probability: 0.481944116123

Try Again