

AutoAI Lab

Introduction

This lab will demonstrate the award-winning AutoAI capability to build and deploy an optimized model based on the Titanic data set.

AutoAI in Cloud Pak for Data 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 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.

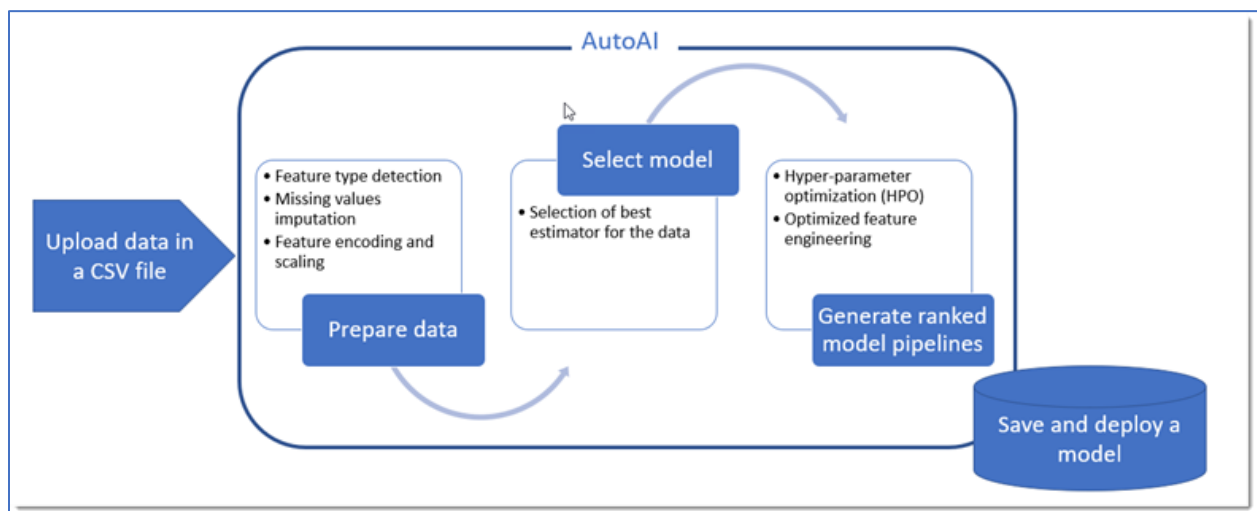


Figure 1- Auto AI Flow

The AutoAI process follows this sequence to build candidate pipelines:

- **Data pre-processing** - Most data sets contain different data formats and missing values, but standard machine learning algorithms work with numbers and no missing values. AutoAI applies various algorithms, or estimators, to analyze, clean, and prepare your raw data for machine learning. It automatically detects and categorizes features based on data type, such as categorical or numerical. Depending on the categorization, it uses hyper-parameter optimization to determine the best combination of strategies for missing value imputation, feature encoding, and feature scaling for your data.
- **Automated model selection** - The next step is automated model selection that matches your data. AutoAI uses a novel approach that enables testing and ranking candidate algorithms against small subsets of the data, gradually increasing the size of the subset

for the most promising algorithms to arrive at the best match. This approach saves time without sacrificing performance. It enables ranking a large number of candidate algorithms and selecting the best match for the data.

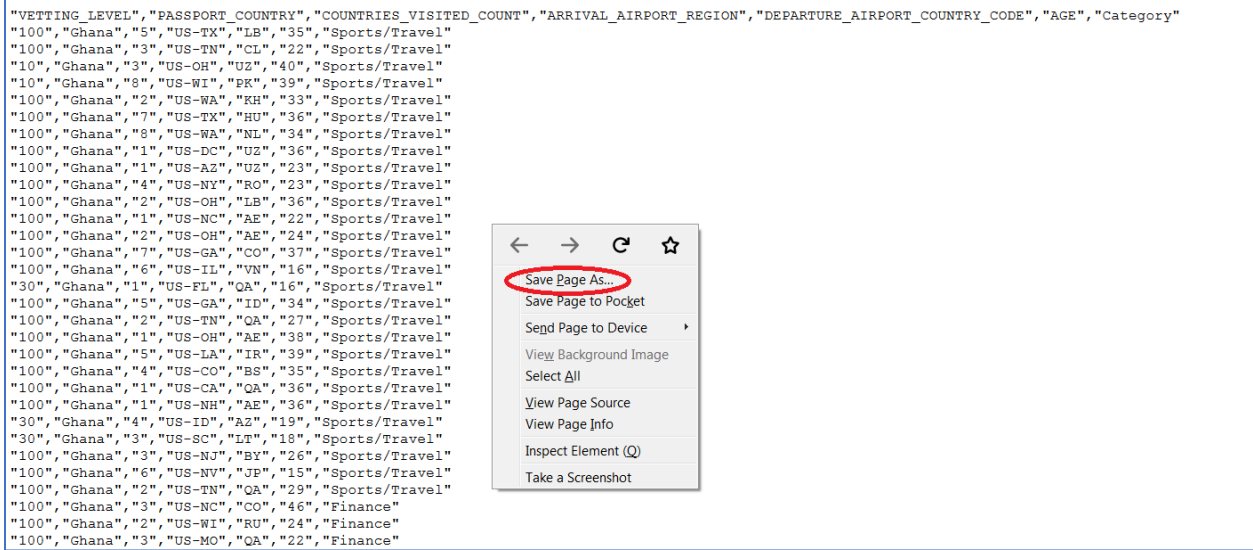
- **Hyperparameter optimization** - Hyper-parameter optimization refines the best performing model pipelines. AutoAI uses a novel hyper-parameter optimization algorithm optimized for costly function evaluations such as model training and scoring that are typical in machine learning. This approach enables fast convergence to a good solution despite long evaluation times of each iteration.
- **Automated feature engineering** - Feature engineering attempts to transform the raw data into the combination of features that best represents the problem to achieve the most accurate prediction. AutoAI uses a unique approach that explores various feature construction choices in a structured, non-exhaustive manner, while progressively maximizing model accuracy using reinforcement learning. This results in an optimized sequence of transformations for the data that best match the algorithms of the model selection step.
- **Repeat Hyperparameter optimization** – The Hyperparameter optimization step is repeated including the derived features from the feature engineering step.

We will perform the following steps in this lab:

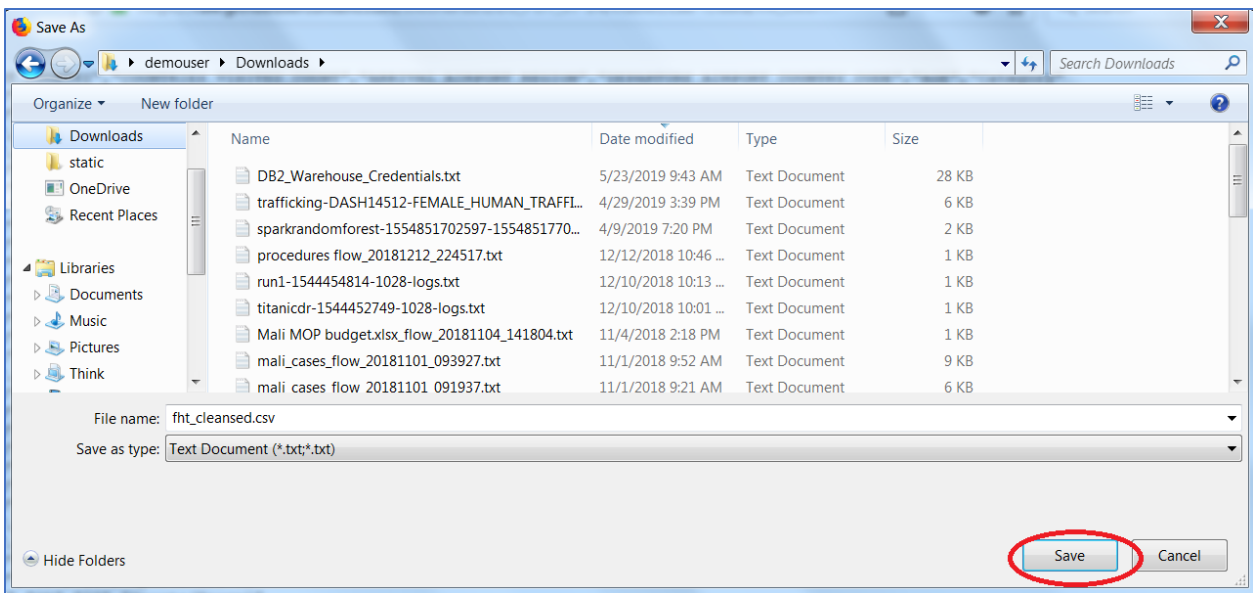
1. Download a female human trafficking cleansed dataset from the github repo
2. Add an Auto AI Experiment to create a model to predict the trafficking risk
3. Save and Deploy the model
4. Test the model

Step 1: Download a female human trafficking cleansed dataset

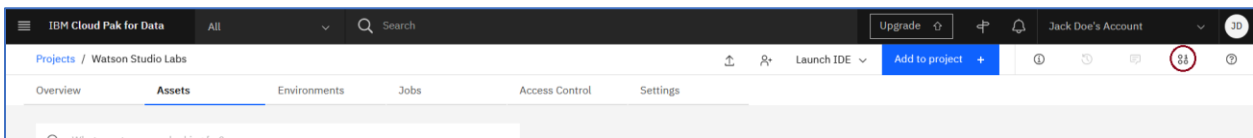
1. Download the fht_cleansed.csv data file from the following location by clicking [here](#)
2. Right-click on the window, and click **Save Page As...**



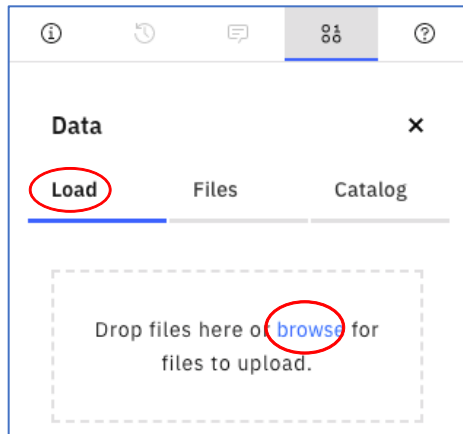
3. Click on Save.



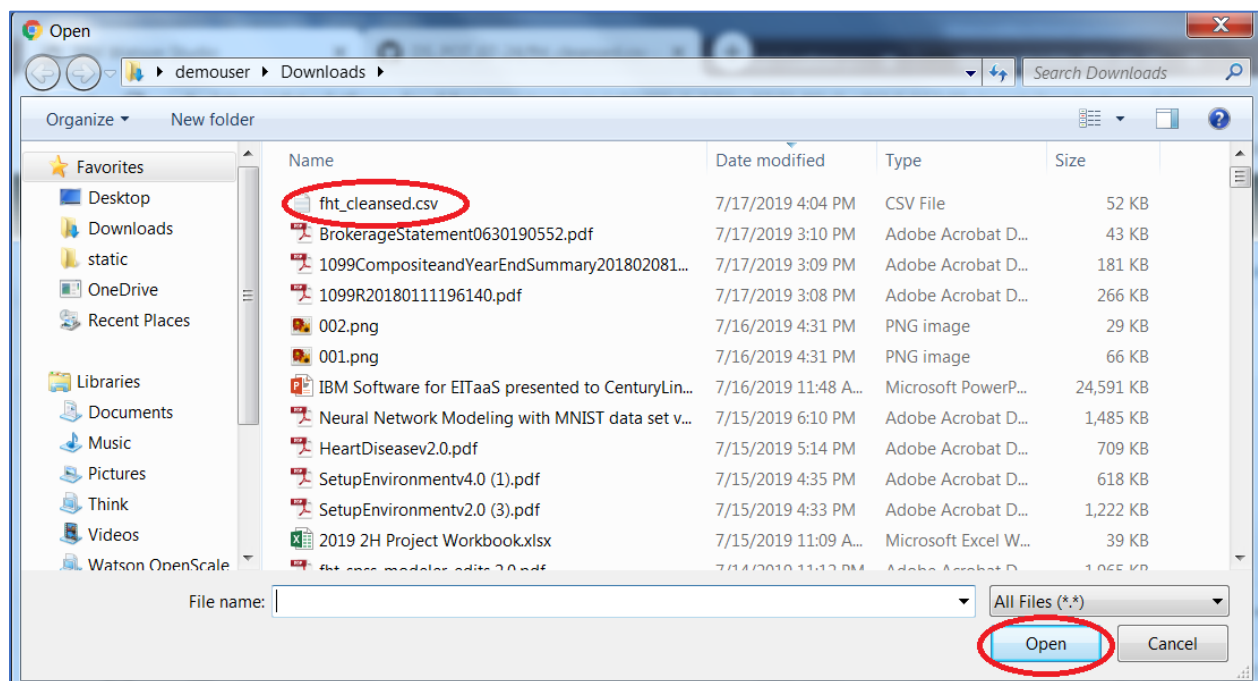
4. Go back to your project. Click on the icon.



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 fht_cleansed.csv file is stored. Select the fht_cleansed.csv file and then click **Open**.

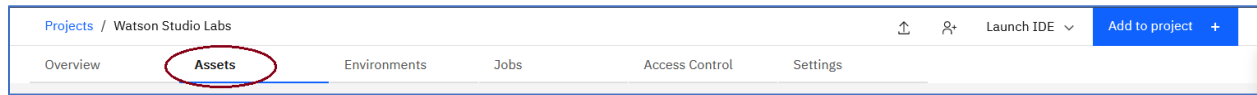


7. The fht_cleansed.csv file is now added as a Data Asset.

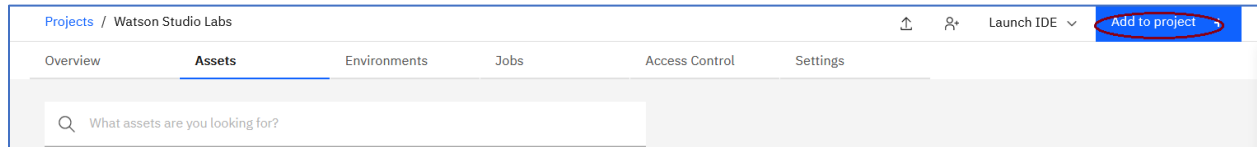
Data assets				
0 assets selected.				
<input type="checkbox"/>	Name	Type	Created by	Last modified
<input type="checkbox"/>	CSV fht_cleansed.csv	Data Asset	FCOT Labs	Aug 03, 2020, 10:18 AM

Step 2: Add an AutoAI Experiment.

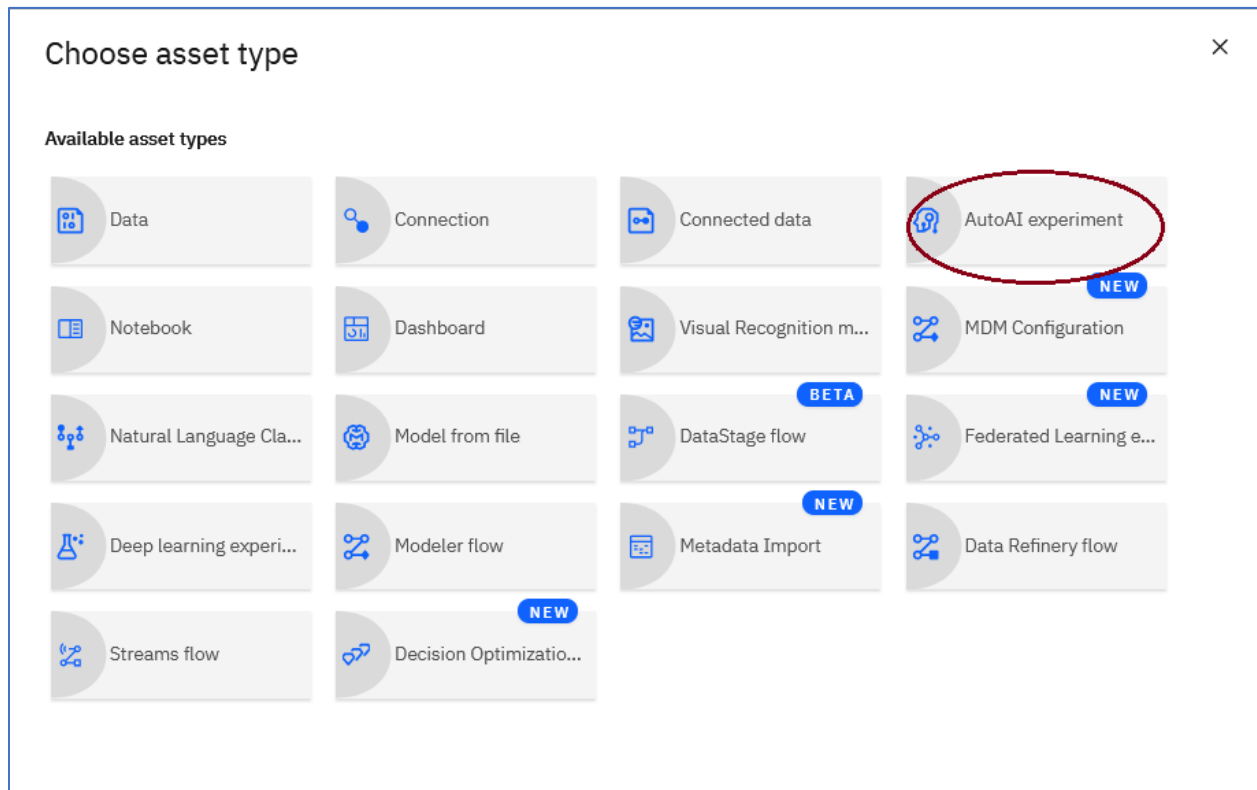
1. If not on the **Assets** page, click on the **Assets** Tab



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



3. Click on **AutoAI Experiment**.



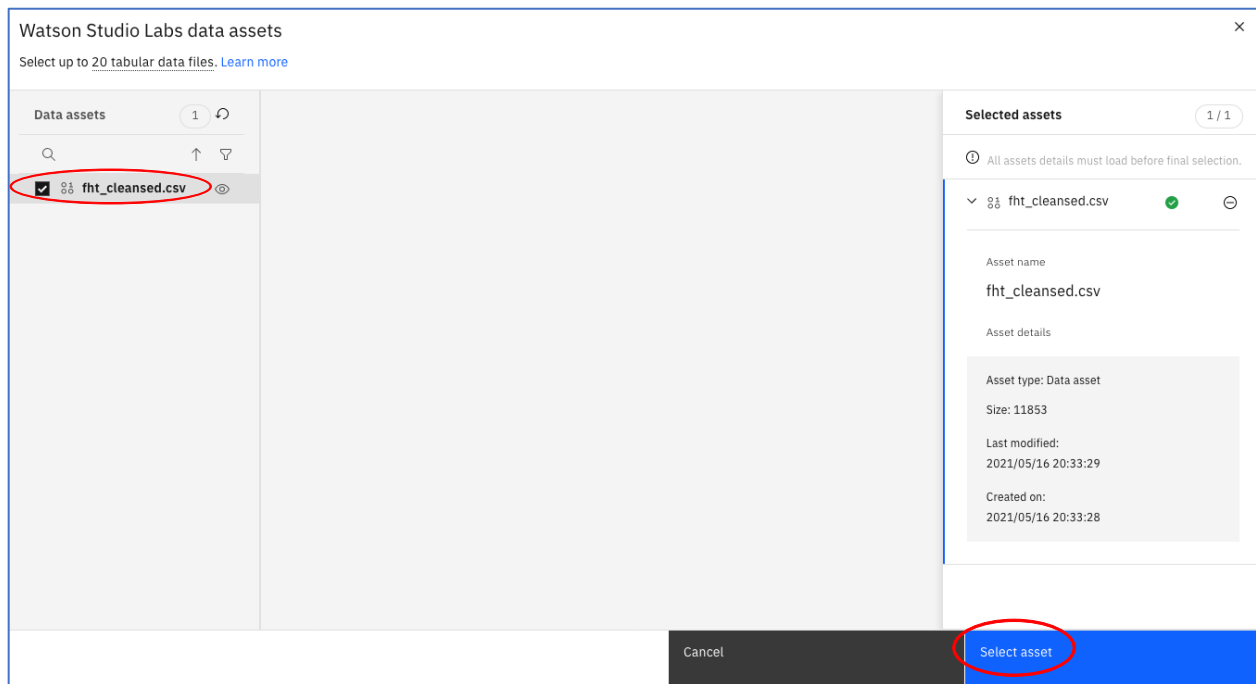
4. Enter an **Asset name**, leave the defaults for the **Watson Machine Learning** and **Compute configuration** and click on **Create**.

The screenshot shows the 'Define details' section of the Watson Machine Learning console. On the left, there is a sidebar with a '+ New' button and a 'Gallery sample' link. The main area is divided into two columns. The left column, titled 'Define details', contains a 'Name *' field with the text 'FHT AutoAI' entered and circled in red, and a 'Description' field with the placeholder text 'Description of AutoAI experiment'. The right column, titled 'Associate services', contains a 'Watson Machine Learning Service Instance *' dropdown menu with 'WatsonMachineLearning' selected, and a 'Compute configuration *' dropdown menu with '8 vCPU and 32 GB RAM' selected. Below these dropdowns, there is a note: 'This compute configuration consumes 20 capacity units per hour. [Learn more](#) about capacity unit hours and Watson Machine Learning pricing plans.' At the bottom right of the form, there are two buttons: 'Cancel' and 'Create', with the 'Create' button circled in red.

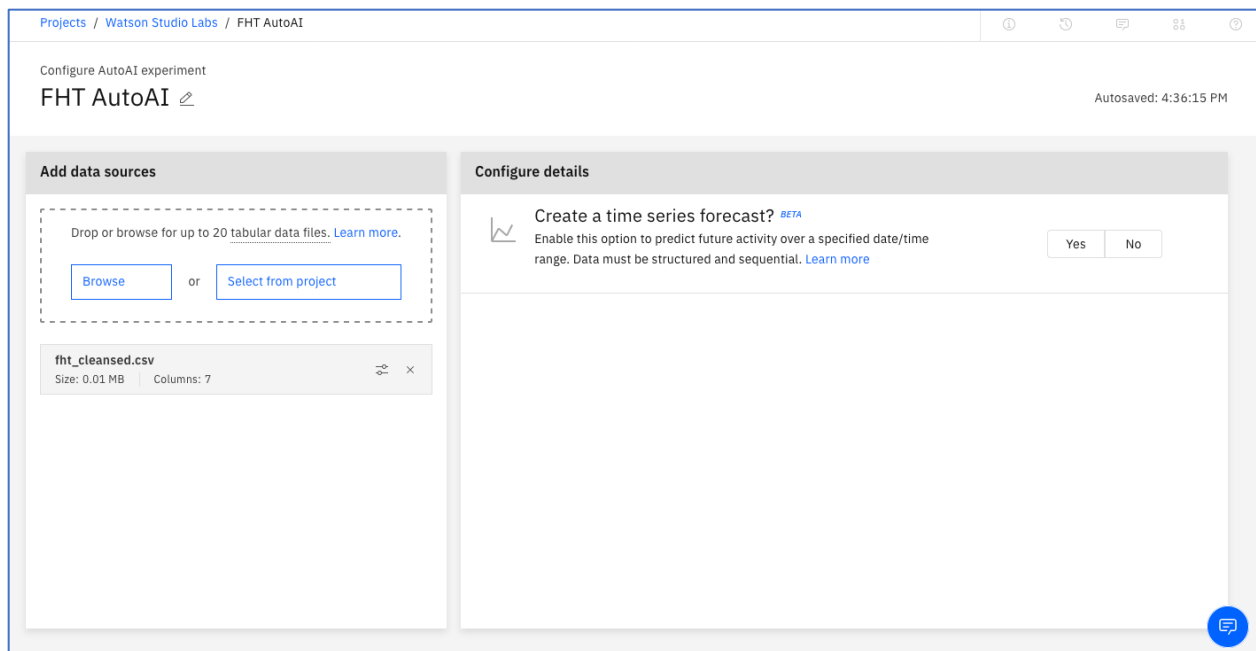
5. Click on **Select from project**.

The screenshot shows the 'Add data source' dialog in the Watson Machine Learning console. The dialog has a title bar that says 'FHT AutoAI' with an edit icon. Below the title bar, there is a section titled 'Add data source' with a dashed border. Inside this section, there is a large empty area for dropping a file. To the right of this area, there is a circular icon with a plus sign and a document. Below the icon, there is the text 'Add data source' and 'Drop or browse for a csv file.' Below this text, there are two buttons: 'Browse' and 'Select from project', with the 'Select from project' button circled in red.

6. Select the **fht_cleansed.csv** dataset and click on **Select asset**.



7. Click **No** next to Create a time series forecast?



8. 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**.

Configure details

Create a time series forecast? BETA
Enable this option to predict future activity over a specified date/time range. Data must be structured and sequential. [Learn more](#)

What do you want to predict?
Prediction column ⓘ
VETTING_LEVEL × ▾

Prediction column: VETTING_LEVEL CUH remaining: 20 CUH

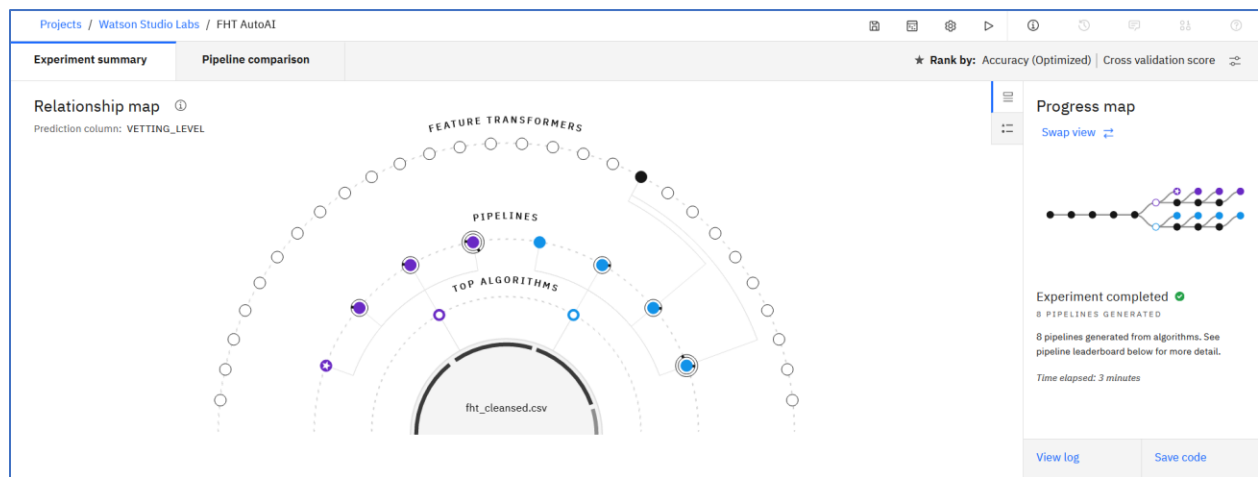
PREDICTION TYPE
Multiclass Classification



OPTIMIZED FOR
Accuracy & run time

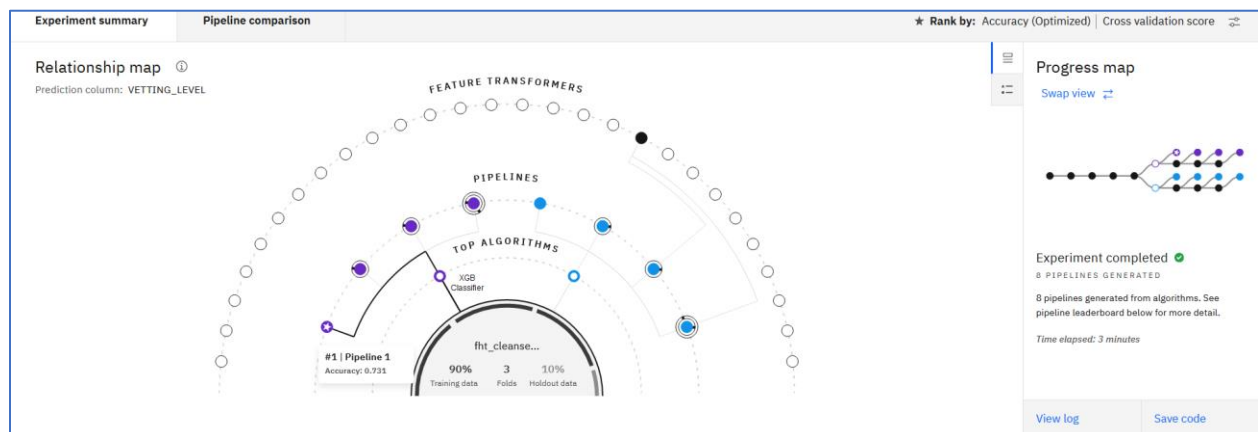
Experiment settings **Run experiment**

9. **Note, your results may not match what is showing below. These results are based on an earlier point in time, and we have a small training set.** It will take several minutes for the eight alternative pipelines to be analyzed. The first pipeline picks the best algorithm. In this case, it is a **XGB** classifier. The second pipeline performs a hyperparameter optimization to see if tuning the algorithm parameters will improve the performance metric. The third pipeline will derive new features (i.e. feature engineering) to try to improve the performance metric. The fourth pipeline will do another hyperparameter optimization including the newly derived features. The next 4 pipelines do the same thing for the second-best algorithm. Note, you can move ahead at any point after the first pipeline has been created. We are going to proceed as if Pipeline 1 is the best ranked pipeline. **Yours could be different.**

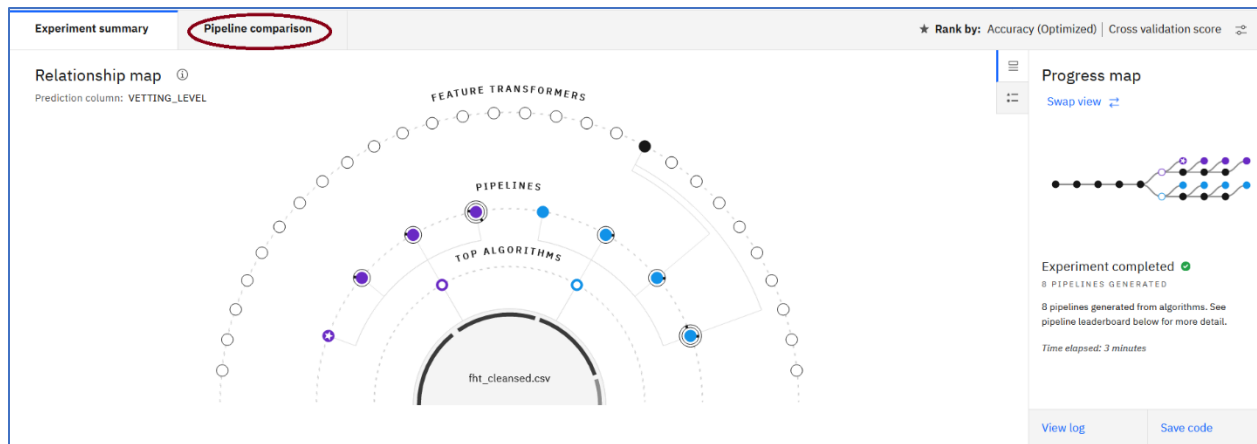
10. When the AutoAI run is completed you will see a display similar to below.



11. Hover over the  icon in the Pipelines semi-circle. This is the top ranked pipeline based on the accuracy metric. It is shown to be Pipeline 1 with a XGB classifier and an accuracy of .73. Note, if you click on the  icon, you will be taken to the metrics for Pipeline 1 (don't do this yet).



12. Click on the **Pipeline comparison**.




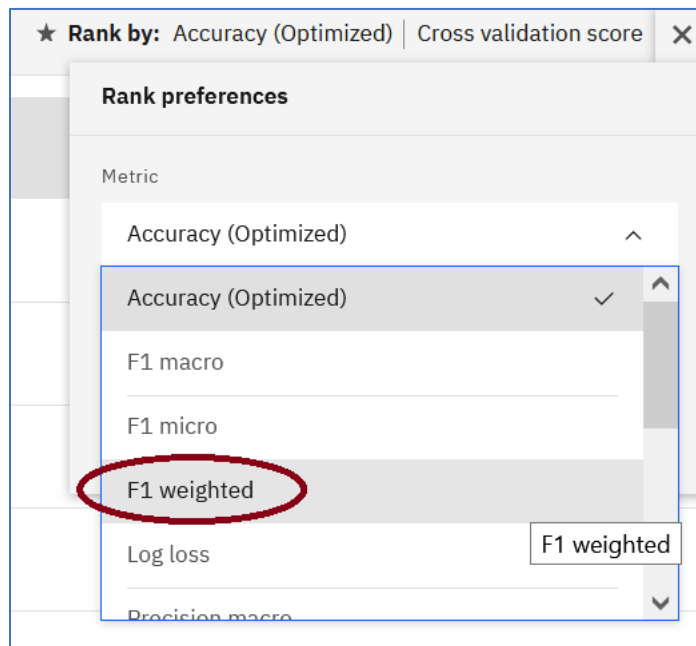
13. This will display a diagram showing the metrics associated with each pipeline on a single chart. Click on **Experiment summary** to return to the prior panel.



14. Scroll down to view the Pipeline Leaderboard ranked by accuracy.


	Rank	↑	Name	Algorithm	Accuracy (Optimized) Cross Validation	Enhancements	Build time
★	1		Pipeline 1	XGB Classifier	0.731	None	00:00:01
	2		Pipeline 2	XGB Classifier	0.731	HPO-1	00:00:09
	3		Pipeline 3	XGB Classifier	0.731	HPO-1 FE	00:00:28
	4		Pipeline 4	XGB Classifier	0.731	HPO-1 FE HPO-2	00:00:20
	5		Pipeline 7	Decision Tree Classifier	0.731	HPO-1 FE	00:00:22
	6		Pipeline 8	Decision Tree Classifier	0.731	HPO-1 FE HPO-2	00:00:09
	7		Pipeline 6	Decision Tree Classifier	0.703	HPO-1	00:00:03
	8		Pipeline 5	Decision Tree Classifier	0.694	None	00:00:01

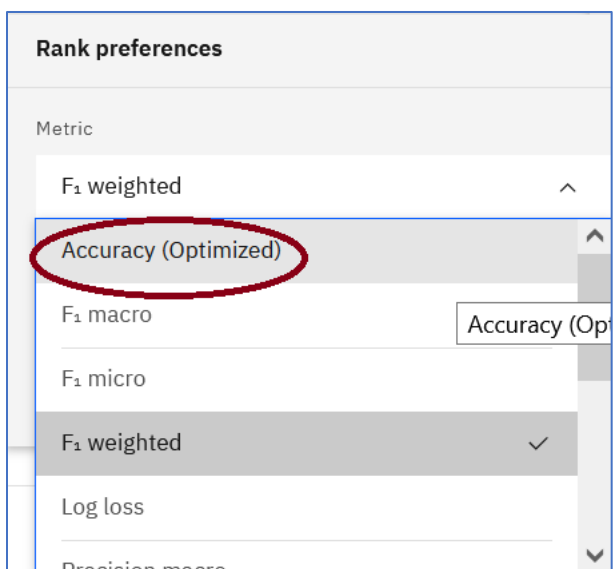
15. Note you can change the optimization metric or whether the metrics apply to the Cross validation sample or the Holdout sample by clicking on the  icon. Click on **F1 Weighted**.



16. The list is now ranked by the F1 Weighted metric.

Experiment summary			Pipeline comparison		★ Rank by: F1 weighted Cross validation score			
	Rank	↑	Name	Algorithm	Accuracy (Optimized) Cross Validation	F1 weighted Cross Validation	Enhancements	Build time
★	1		Pipeline 7	Decision Tree Classifier	0.731	0.731	HPO-1 FE	00:00:22
	2		Pipeline 8	Decision Tree Classifier	0.731	0.731	HPO-1 FE HPO-2	00:00:09
	3		Pipeline 1	XGB Classifier	0.731	0.731	None	00:00:01
	4		Pipeline 2	XGB Classifier	0.731	0.731	HPO-1	00:00:09
	5		Pipeline 3	XGB Classifier	0.731	0.731	HPO-1 FE	00:00:28
	6		Pipeline 4	XGB Classifier	0.731	0.731	HPO-1 FE HPO-2	00:00:20
	7		Pipeline 6	Decision Tree Classifier	0.703	0.698	HPO-1	00:00:03
	8		Pipeline 5	Decision Tree Classifier	0.694	0.695	None	00:00:01

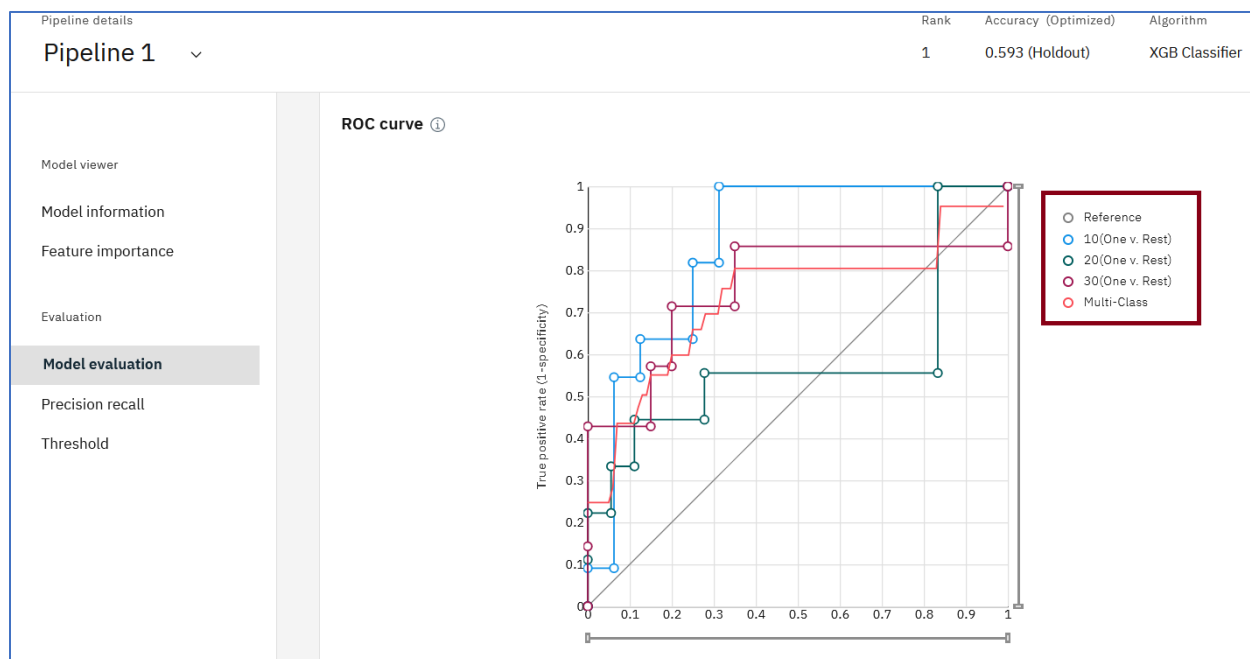
17. Return to the Accuracy ranked list by clicking on the  and selecting **Accuracy**



18. Click on the highest ranked pipeline (**pipeline 1 in the picture**).

Experiment summary		Pipeline comparison		★ Rank by: Accuracy (Optimized) Cross validation score ↕			
	Rank ↑	Name	Algorithm	Accuracy (Optimized) Cross Validation	F1 weighted Cross Validation	Enhancements	Build time
★	1	Pipeline 1	XGB Classifier	0.731	0.731	None	00:00:01
	2	Pipeline 2	XGB Classifier	0.731	0.731	HPO-1	00:00:09
	3	Pipeline 3	XGB Classifier	0.731	0.731	HPO-1 FE	00:00:28

19. The **Model Evaluation** panel is displayed. The model evaluation metric for the holdout sample is displayed. The Receiver Operating Characteristic (ROC) plots the true positive rate versus the false positive rate at varying thresholds. Highlighted in the maroon rectangle is a clickable legend. Clicking on the circles will toggle the corresponding display on/off.



20. Scroll down to view model metrics for both the training sample and the holdout sample.

View

Multi-Class

Measures	Holdout score	Cross validation score
Precision macro	0.624	0.734
Accuracy	0.593	0.731
Recall macro	0.592	0.721
Weighted precision	0.639	0.738
F1 macro	0.571	0.724
Weighted f1 measure	0.581	0.731
Weighted recall	0.593	0.731
Log loss	1.653	0.848

21. Click on **Feature Summary**.

Pipeline details

Pipeline 1

Model viewer

Model information

Feature summary

Evaluation

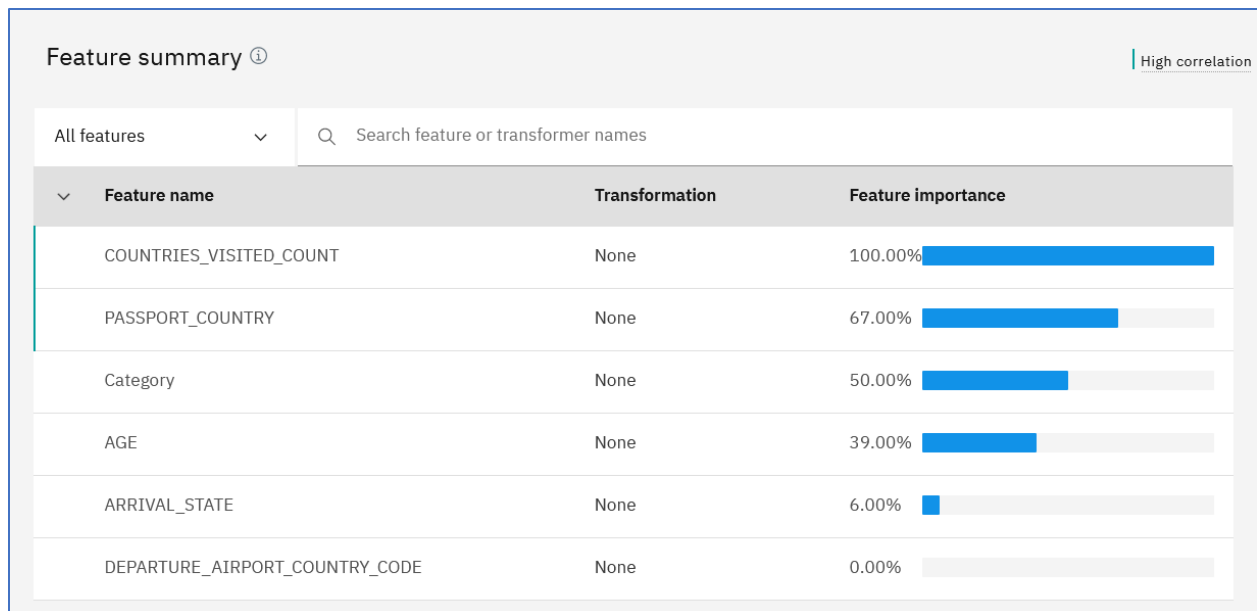
Model evaluation

Confusion matrix

Precision recall

Threshold

22. According to the **Feature Importance**, the **COUNTRIES_VISITED_COUNT** feature is the most important.



Step 3: Save and Deploy the Model

In this section, we will save and then deploy the model. You create a deployment for a machine learning model so you can submit new data to a model and get a score, or prediction back. To

deploy the model, you must have a *deployment space* where you can organize the assets you need to create and monitor deployments. A space contains an overview of deployment status, the deployable assets, deployments, associated input and output data, and the associated environments.

To deploy a model, we first **promote** the model to a deployment space. When you promote a model to a space, components required for a successful deployment, such as a training library, model definition, other dependent assets, or environment definition are automatically promoted as well. After promoting a model and the data assets needed to a deployment space, you can create a deployment in the space.

You can create a deployment space, promote the model, and create a deployment through the user interface or programmatically.

1. Click on Save as.

Pipeline details	Rank	Accuracy (Optimized)	Algorithm	Enhancements	
Pipeline 1	1	0.593 (Holdout)	XGB Classifier	None	Save as

- AutoAI can generate both a machine learning model as well as a Jupyter notebook. The Jupyter notebook can contain the source code to build the model, or the source code plus the deployment code. Feel free to experiment with saving the notebooks. For this lab, we will save the model. Make sure the Model is checked, optionally change the **Name**, and click **Create**.

Select asset type

Model

Create a Watson Machine Learning model asset that you can test with new data, deploy to generate predictions, and trace lineage activity.

Notebook

Create a notebook if you want to view the code that created this model pipeline or interact with the model programmatically.

Define details

Name

FHT AutoAI - P1 XGB Classifier

Description (optional)

Enter description here

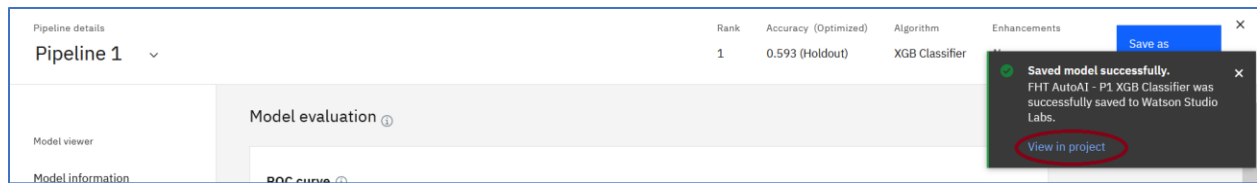
Tags

Add tags to make assets easier to find.

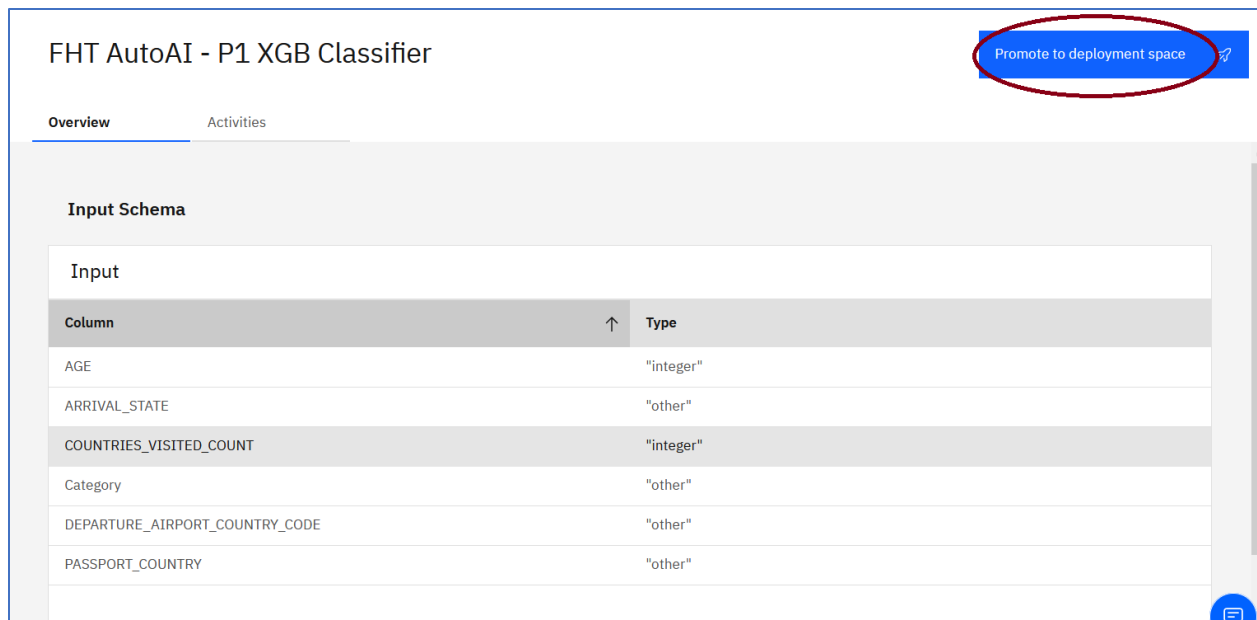
Add a tag

[Cancel](#)[Create](#)

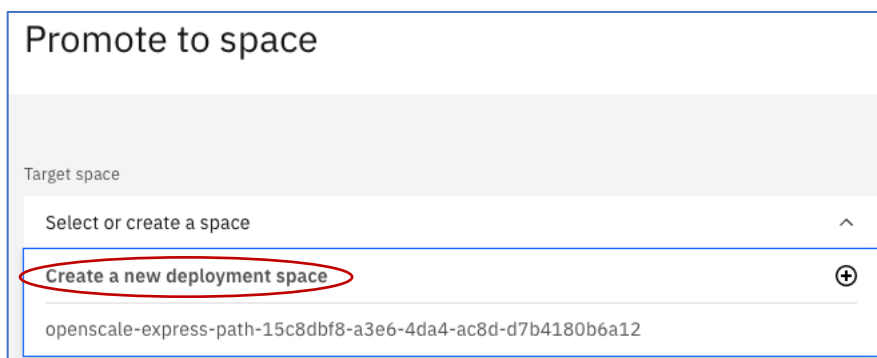
- A message is displayed showing the model was successfully saved. Click on **View in project**.



4. As mentioned above, we will need to promote the model to a deployment space. Click on **Promote to deployment space**.



5. We will create a new space for our model deployment. Click on **New space**.



6. Enter **AutoAI Deployment Space** for the Name, scroll down and select **WatsonMachineLearning** for the **Select machine learning service**, and click **Create**.

Create a deployment space

Use a space to collect assets in one place to create, run, and manage deployments

Name

Description (Optional)

Deployment space tags (optional) ⓘ

Select storage service ⓘ

Select machine learning service (optional) ⓘ

7. Two message boxes are displayed. Click **Close** in the second message box.

The space is being prepared...

The space "AutoAI Deployment Space" is being created.

○ Step 1 of 1. Creating deployment space.

The space is ready

Import is complete! Click **View new space** to view the space and associated assets.

✓ Step 1 of 1. Creating deployment space.

8. Click **Promote**.

Target space
AutoAI Deployment Space

Tags (optional)
Start typing to add tags

Why don't I see all of my spaces? ⓘ
☐ Go to the model in the space after promoting it

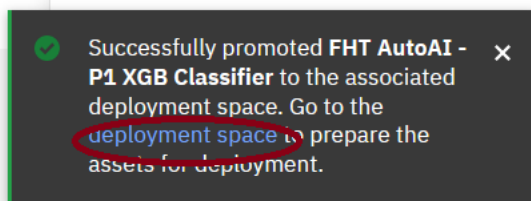
Selected assets (1)



Asset name	Format
FHT AutoAI - P1 XGB Classifier	Model

Description (optional)
Description of assets

Cancel Promote

9. A message is displayed. Click **Deployment Space** in the message.






10. Hover over the promoted model until you see the deploy icon  appear on the right. Click on the  icon.

AutoAI Deployment Space

Overview **Assets** Deployments Jobs Manage

Q What assets are you looking for?

Models (1) [Import model +](#)

Name	Type	Software specification	Tags	Last modified	
 FHT AutoAI - P1 XGB Classifier	wml-hybrid_0.1	hybrid_0.1		Aug 24, 2021 11:40 AM	 


Deploy

11. Click **Online** for the **Deployment type**, enter **FHT AutoAI Deployment** for the **Name**, and click **Create**.

Create a deployment

Associated asset
FHT AutoAI - P1 XGB Classifier

Deployment type

Online  **Batch**

Run the model on data in real-time, as data is received by a web service.

Run the model against data as a batch process.

Name
FHT AutoAI Deployment

Serving name ⓘ
Deployment serving name

Cancel **Create**

12. Click the Deployments tab. Once the asset has “Deployed” under Status, click on the deployment name. You can click the browser refresh if the status doesn’t change after 1 minute or so.

Deployments / AutoAI Deployment Space ↑ Add to space +

AutoAI Deployment Space

Assets **Deployments** Jobs Manage

Q What deployments are you looking for?

Deployments (1)

Name	Type	Status	Asset	Tags	Last modified	↓
FHT AutoAI Deployment	Online	Deployed	FHT AutoAI - P2 XGBClassifier		May 16, 2021 6:42 PM	

13. The **API Reference** 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. Click the **Test** tab.

FHT AutoAI Deployment Deployed Online

API reference **Test**

Direct link

Endpoint Bearer <token> ⓘ

`https://us-south.ml.cloud.ibm.com/ml/v4/deployments/75a93d69-f6ac-4b8a-b4d8-43b70d38e411/p...` Ⓜ IAM

Code snippets

cURL Java JavaScript Python Scala

```
# NOTE: you must set $API_KEY below using information retrieved from your IBM Cloud account.
curl --insecure -X POST --header "Content-Type: application/x-www-form-urlencoded" --header "Accept: application/json" --data-urlencode "grant_type=urn:ibm:params:oauth:grant-type=apikey" --url https://us-south.ml.cloud.ibm.com/ml/v4/deployments/75a93d69-f6ac-4b8a-b4d8-43b70d38e411/predict --header "Authorization: Bearer $API_KEY"

# the above CURL request will return an auth token that you will use as $IAM_TOKEN in the scoring request below
# TODO: manually define and pass values to be scored below
curl -X POST --header "Content-Type: application/json" --header "Accept: application/json" --header "Authorization: Bearer $IAM_TOKEN" -d '{"input_data": [{"fields": [{"PASSPORT_COUNTRY": "Ghana", "COUNTRIES_VISITED_COUNT": 3, "ARRIVAL_STATE": "OH", "DEPARTURE_AIRPORT_COUNTRY_CODE": "RU", "AGE": 20, "Category": "Sports/Travel"}]}]}'
```

14. Enter the following values:

PASSPORT_COUNTRY – **Ghana**

COUNTRIES_VISITED_COUNT – **3**

ARRIVAL_STATE – **OH**

DEPARTURE_AIRPORT_COUNTRY_CODE – **RU**

AGE – **20**

Category – **Sports/Travel**

Click **Add to list**.

Deployments / AutoAI Deployment Space / FHT AutoAI - P2 XGBClassifier / FHT AutoAI Deployment

FHT AutoAI Deployment Deployed Online

API reference **Test**

Enter input data

PASSPORT_COUNTRY

Ghana

COUNTRIES_VISITED_COUNT

3

ARRIVAL_STATE

OH

DEPARTURE_AIRPORT_COUNTRY_CODE

RU

AGE

20

Category

Sports/Travel

Add to list

+

Input list (0)

The prediction list is empty.
Add one or more items to the list to request a prediction.

Predict

Result

Submit a prediction to see scoring results.

15. Click Predict (1)

Deployments / AutoAI Deployment Space / FHT AutoAI - P2 XGBClassifier / FHT AutoAI Deployment

FHT AutoAI Deployment Deployed Online

API reference **Test**

Enter input data

PASSPORT_COUNTRY

other

COUNTRIES_VISITED_COUNT

Integer

ARRIVAL_STATE

other

DEPARTURE_AIRPORT_COUNTRY_CODE

other

AGE

Integer

Category

other

Add to list

+

Input list (1)

[Ghana, 3, OH, RU, 20, Sports/Travel]

Predict (1)

Result

Submit a prediction to see scoring results.

16. The prediction for VETTING_LEVEL is **30 (Low Risk)** with **54.3 %** confidence.

The screenshot displays the 'FHT AutoAI Deployment' interface. The top navigation bar shows the path: Deployments / AutoAI Deployment Space / FHT AutoAI - P2 XGBClassifier / FHT AutoAI Deployment. The main header indicates the model is 'Deployed' and 'Online'. Below this, there are tabs for 'API reference' and 'Test', with 'Test' being the active tab.

The 'Test' tab is divided into three main sections:

- Enter input data:** This section contains several input fields for testing the model:
 - PASSPORT_COUNTRY: other
 - COUNTRIES_VISITED_COUNT: Integer
 - ARRIVAL_STATE: other
 - DEPARTURE_AIRPORT_COUNTRY_CODE: other
 - AGE: Integer
 - Category: other
- Input list (1):** This section shows a single input example: [Ghana, 3, OH, RU, 20, Sports/Travel].
- Result:** This section displays the model's output in JSON format. The prediction is 30 (Low Risk) with a confidence of 54.3%. The JSON structure is as follows:

```
0 {  
1   "predictions": [  
2     {  
3       "fields": [  
4         "prediction",  
5         "probability"  
6       ],  
7       "values": [  
8         30,  
9         0.4497275650501251,  
10        0.0672724349498748,  
11        0.5430530905723572  
12      ]  
13    }  
14  ]  
15 }  
16  
17  
18  
19 }
```

You have successfully completed the Lab!!!

- ✓ Downloaded a female human trafficking cleansed dataset from the github repo
- ✓ Added an Auto AI Experiment to create a model to predict the trafficking risk
- ✓ Saved and Deployed the model
- ✓ Tested the model