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.

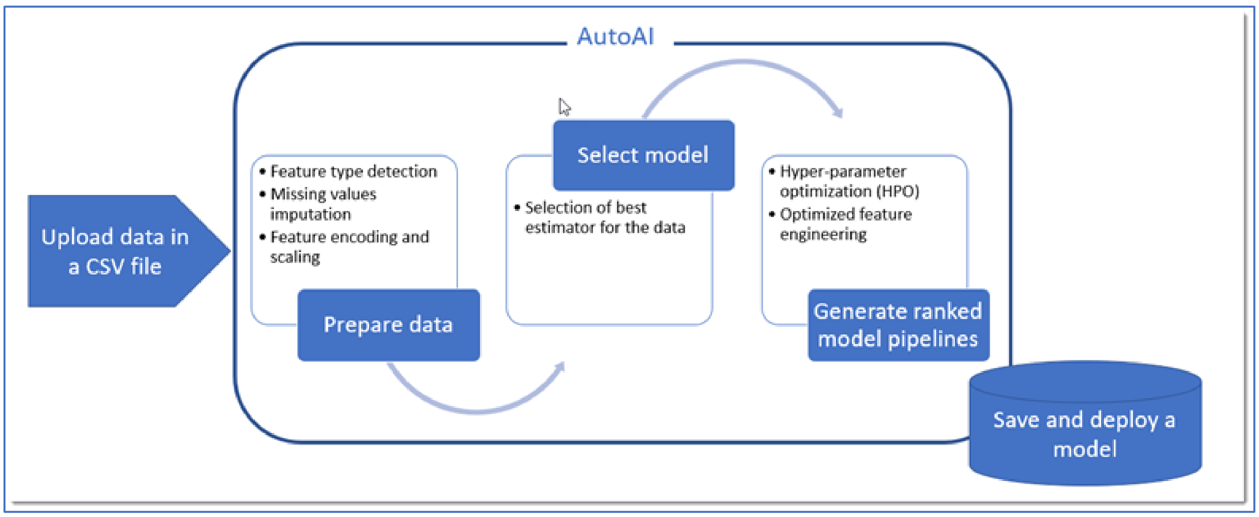


Figure 1- Auto AI Flow

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.

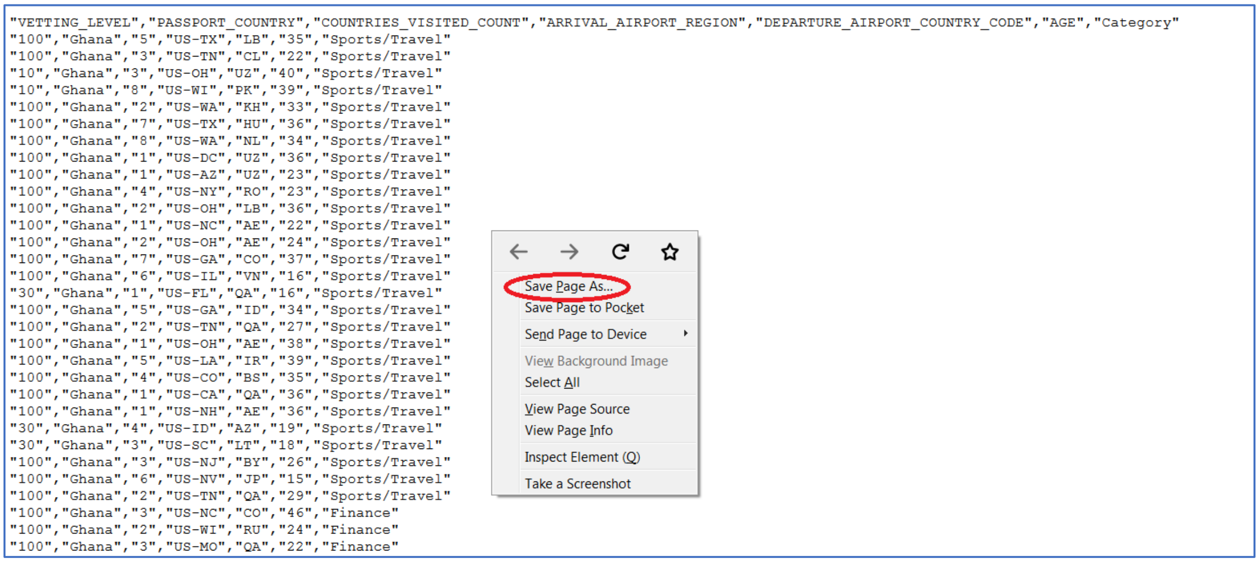
- **Repeat Hyperparameter optimization –** The Hyperparameter optimization step is repeated including the derived features from the feature engineering step.

We will perform the following steps in this lab:

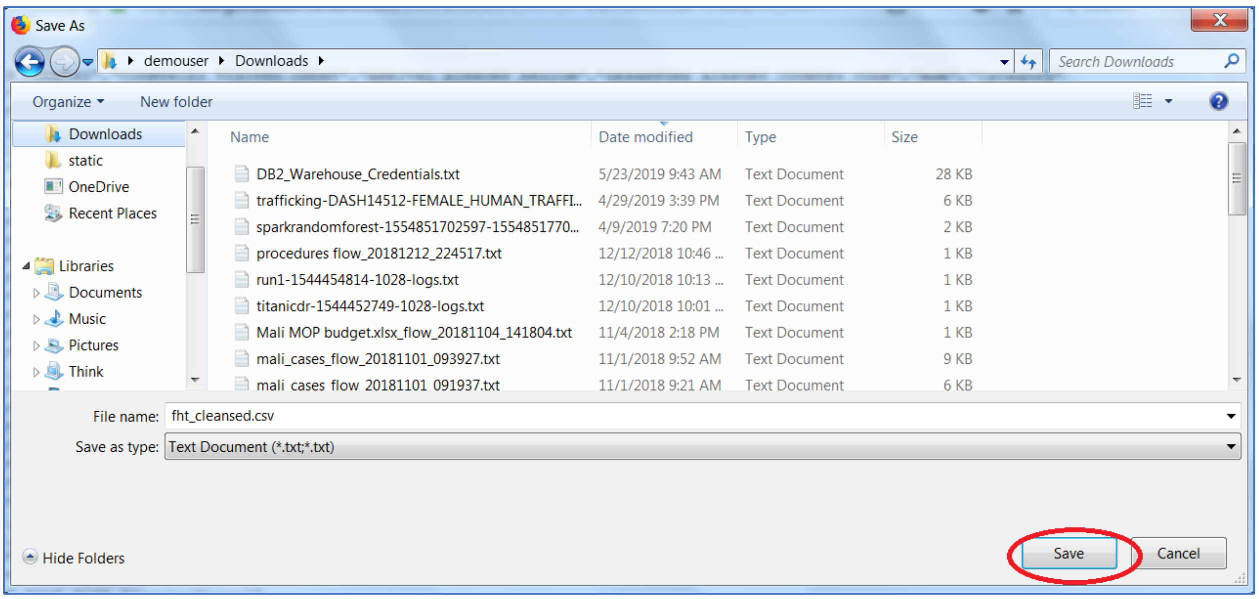
1. Download a female human trafficking cleansed dataset from the github repo
2. Add an Auto AI Experiment to create a model to predict the trafficking risk
3. Save and Deploy the model
4. Test the model

Step 1: Download a female human trafficking cleansed dataset

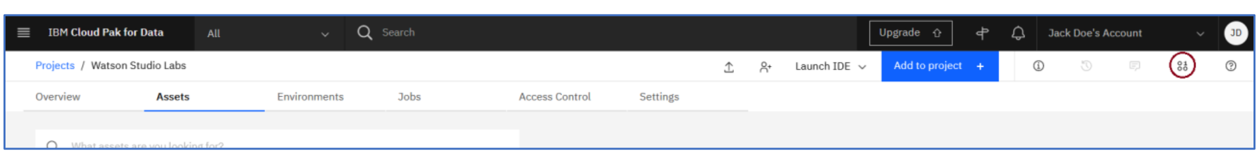
1. Download the fht_cleansed.csv data file from the following location by clicking here
2. Right-click on the window, and click **Save Page As…**



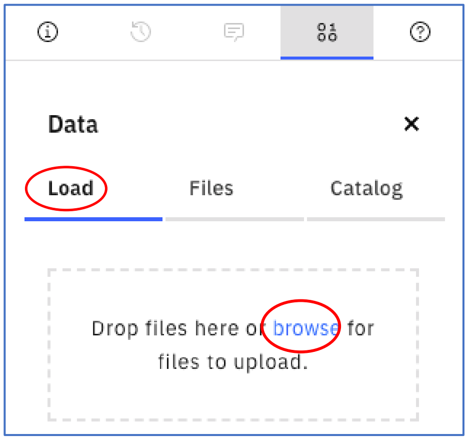
3. Click on **Save**.



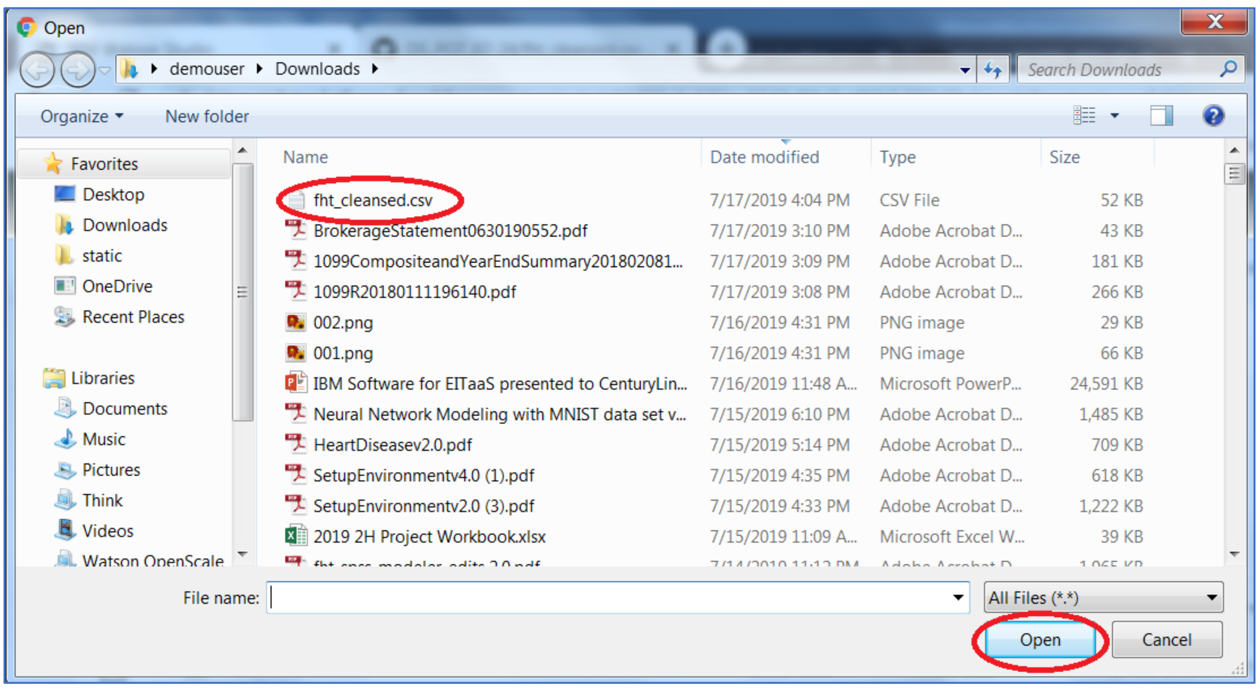
4. Go back to your project. Click on the icon.



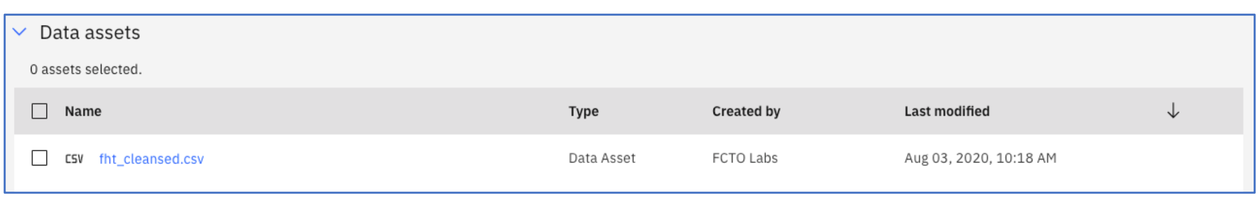
5. Click on the **Load** tab and then click on **browse**. If you don't see the **Load** tab, click on the icon again.



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

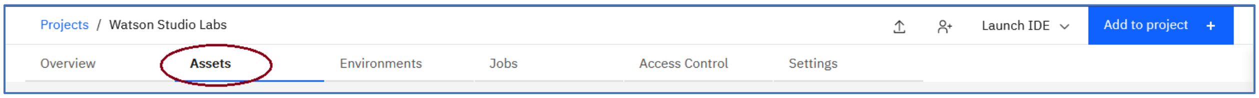


7. The fht_cleansed.csv file is now added as a Data Asset.

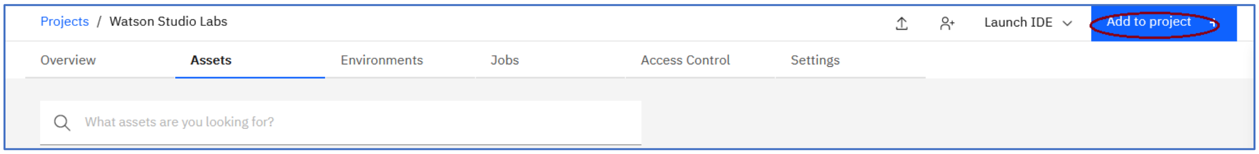


Step 2: Add an AutoAI Experiment.

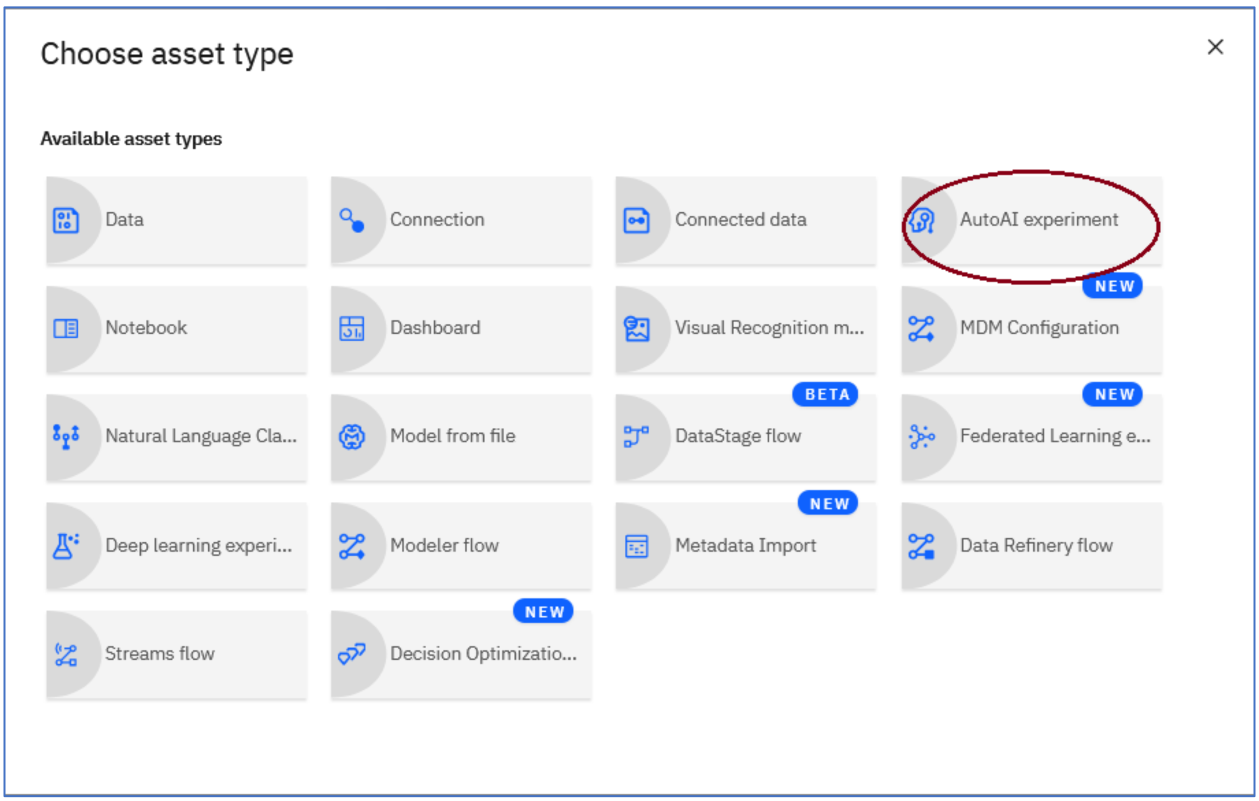
1. If not on the **Assets** page, click on the **Assets** Tab



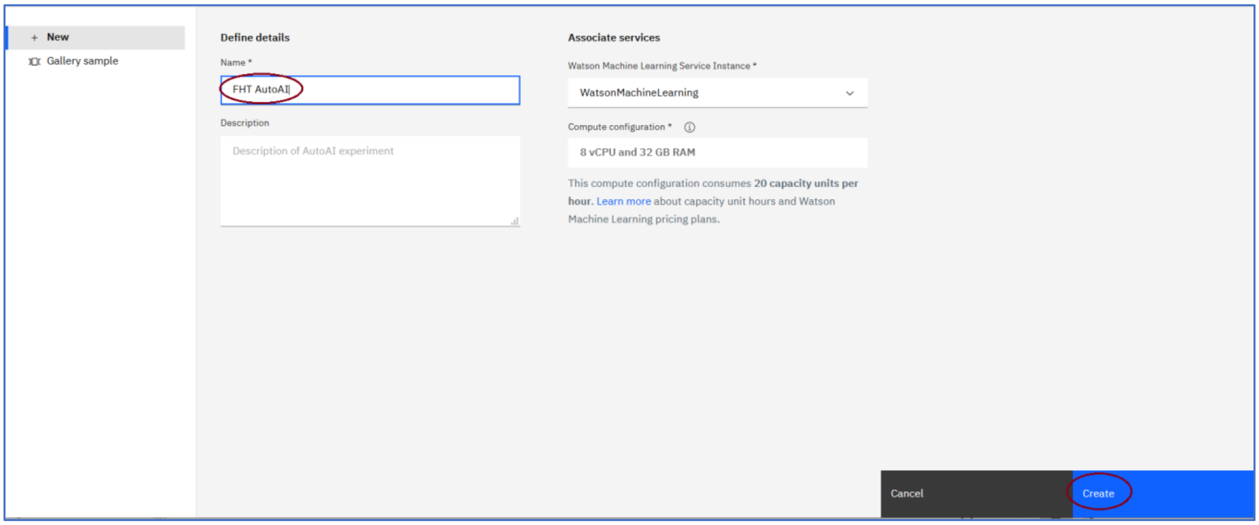
2. Click on **Add to project**.



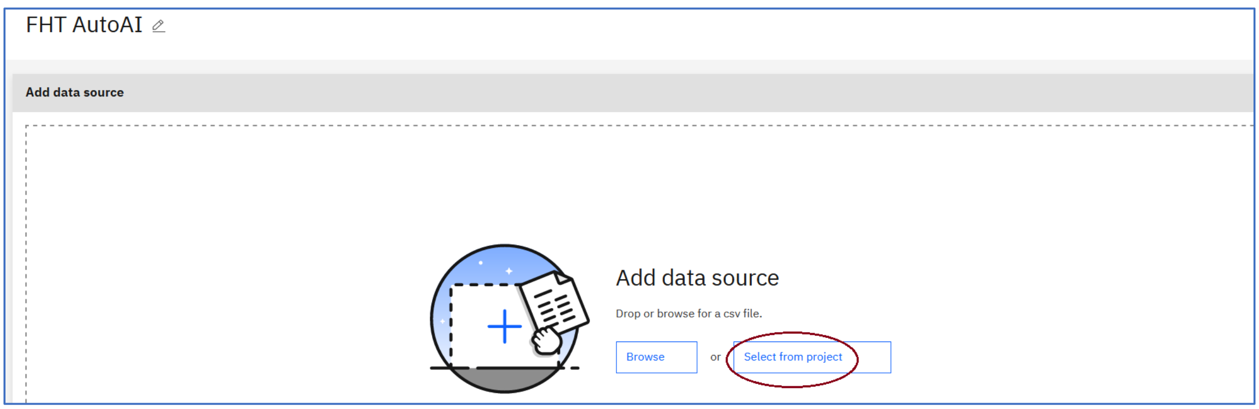
3. Click on **AutoAI Experiment.**



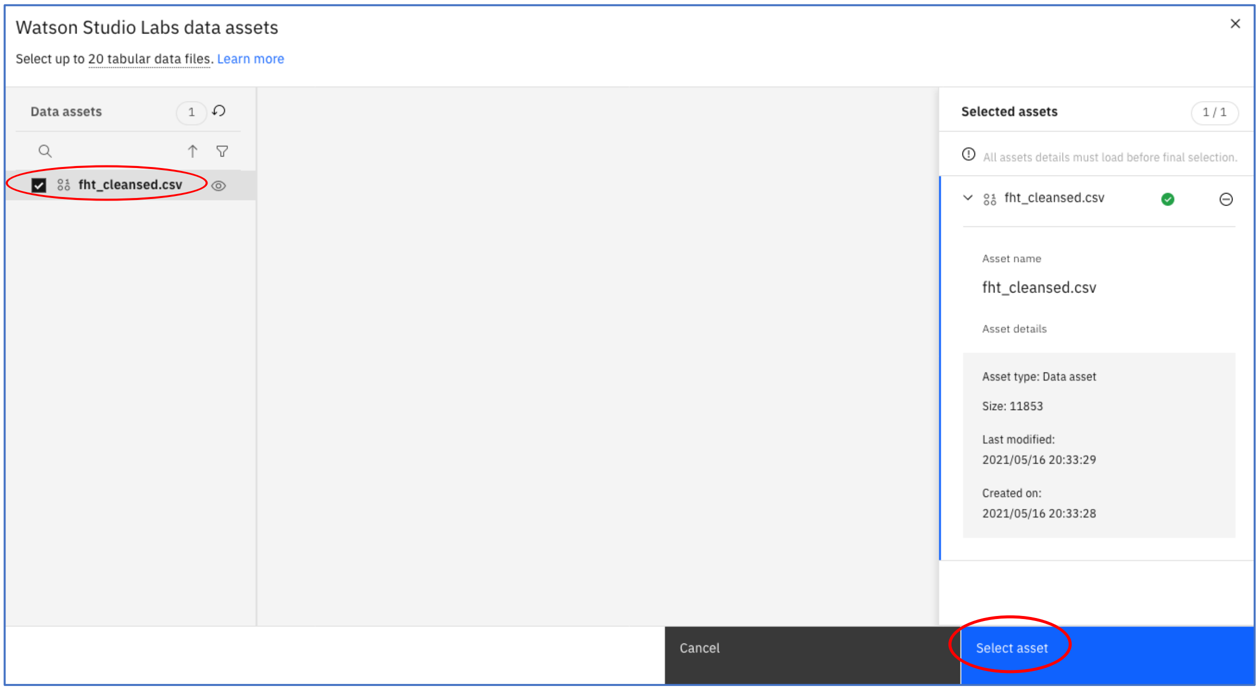
4. Enter an **Asset name**, leave the defaults for the **Watson Machine Learning** and **Compute configuration** and click on **Create**.



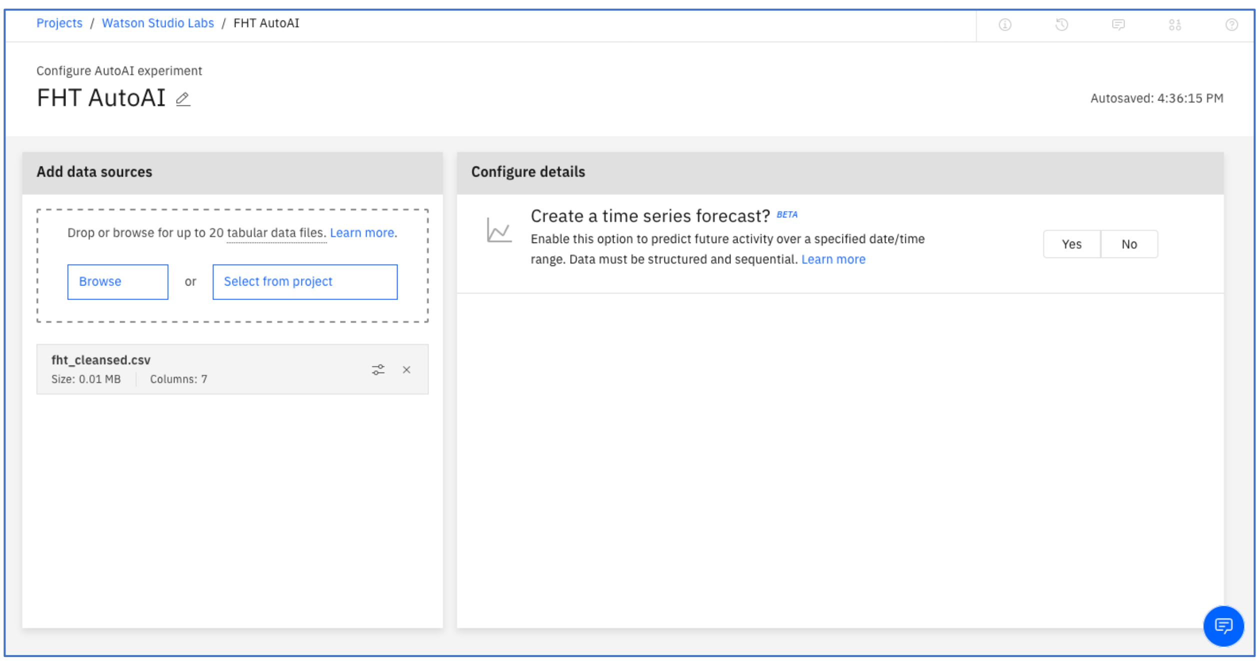
5. Click on **Select from project**.



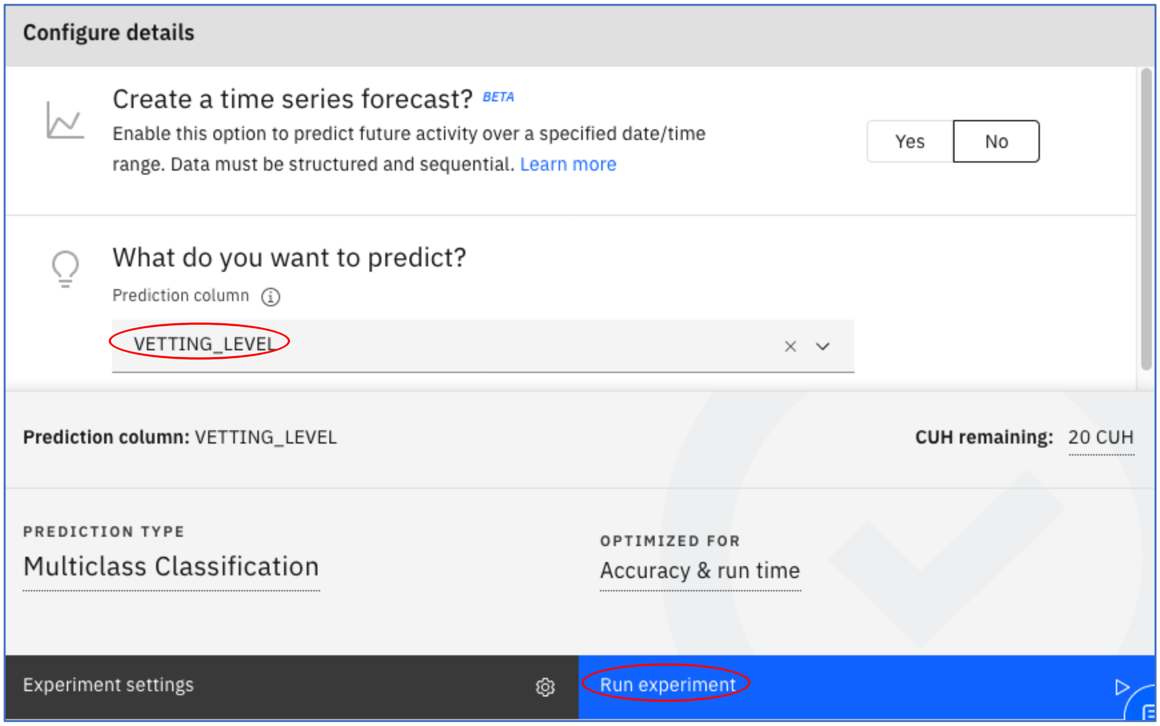
6. Select the **fht_cleansed.csv** dataset and click on **Select asset**.



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

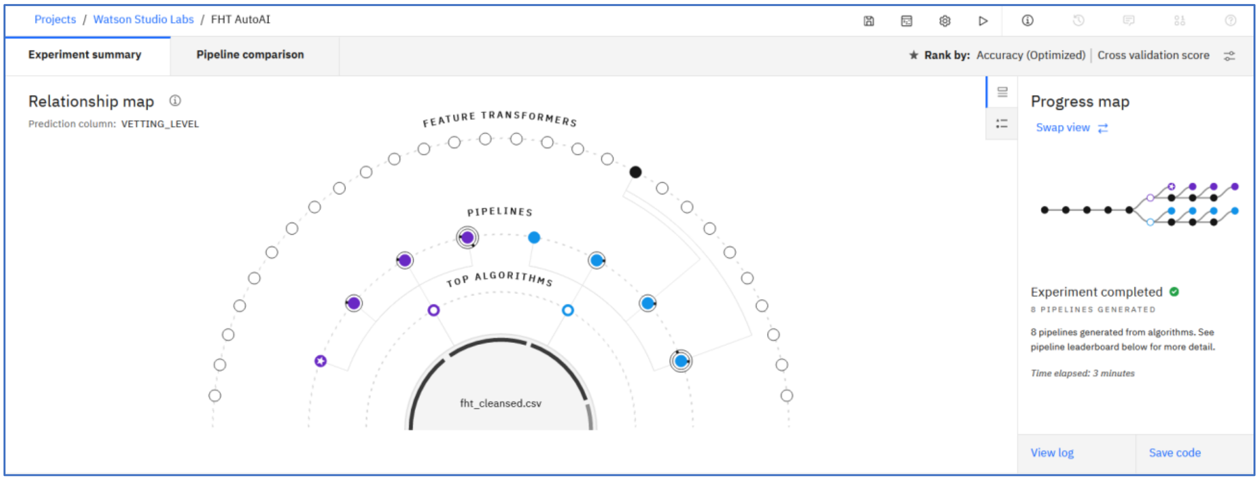


8. Select **VETTING_LEVEL** as the column to predict, leave the default for **OPTIMIZED METRIC** and click on **Run Experiment**. Note the system scanned the VETTING_LEVEL values to determine that a **Multiclass Classification** was the **PREDICTION TYPE**.

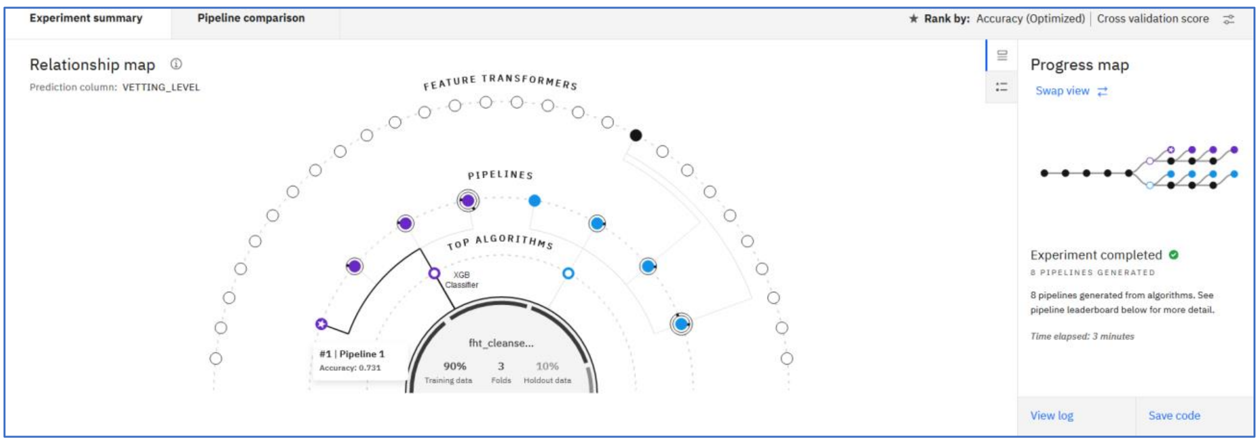


9. **Note, your results may not match what is showing below. These results are based on an earlier point in time, and we have a small training set.** It will take several minutes for the eight alternative pipelines to be analyzed. The first pipeline picks the best algorithm. In this case, it is a **XGB** 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**.

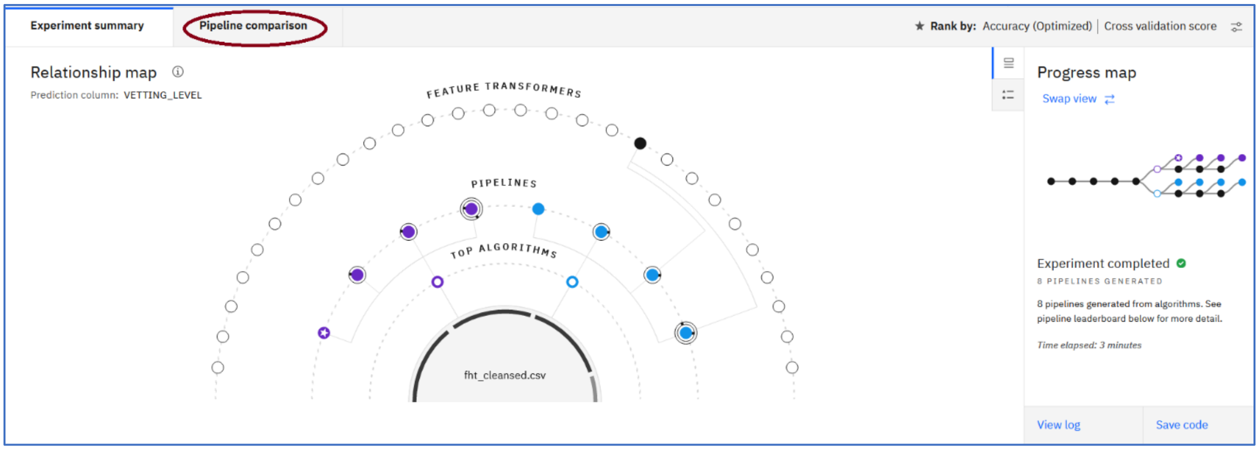
10. When the AutoAI run is completed you will see a display similar to below.



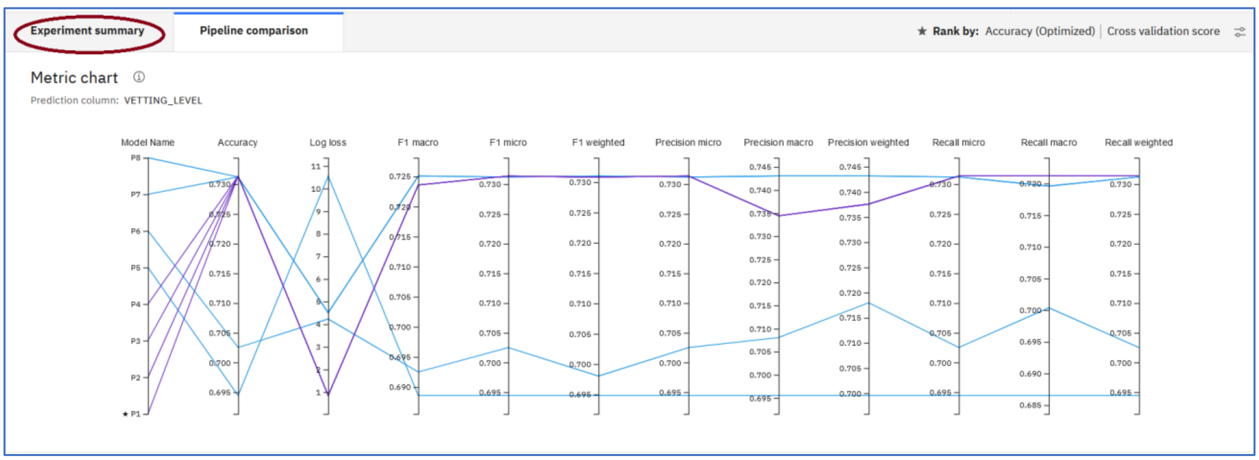
11. Hover over the  icon in the Pipelines semi-circle. This is the top ranked pipeline based on the accuracy metric. It is shown to be Pipeline 1 with a XGB classifier and an accuracy of .73. Note, if you click on the  icon, you will be taken to the metrics for Pipeline 1 (don't do this yet).



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



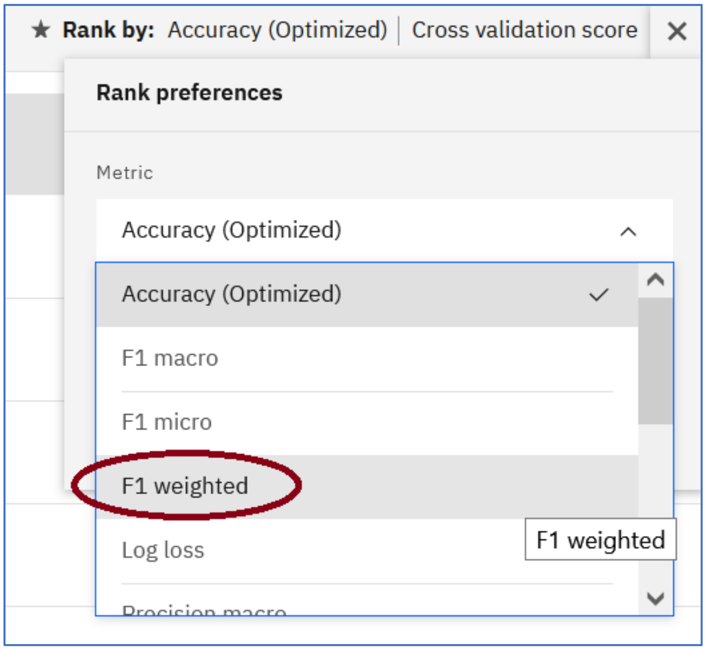
13. This will display a diagram showing the metrics associated with each pipeline on a single chart. Click on **Experiment summary** to return to the prior panel.



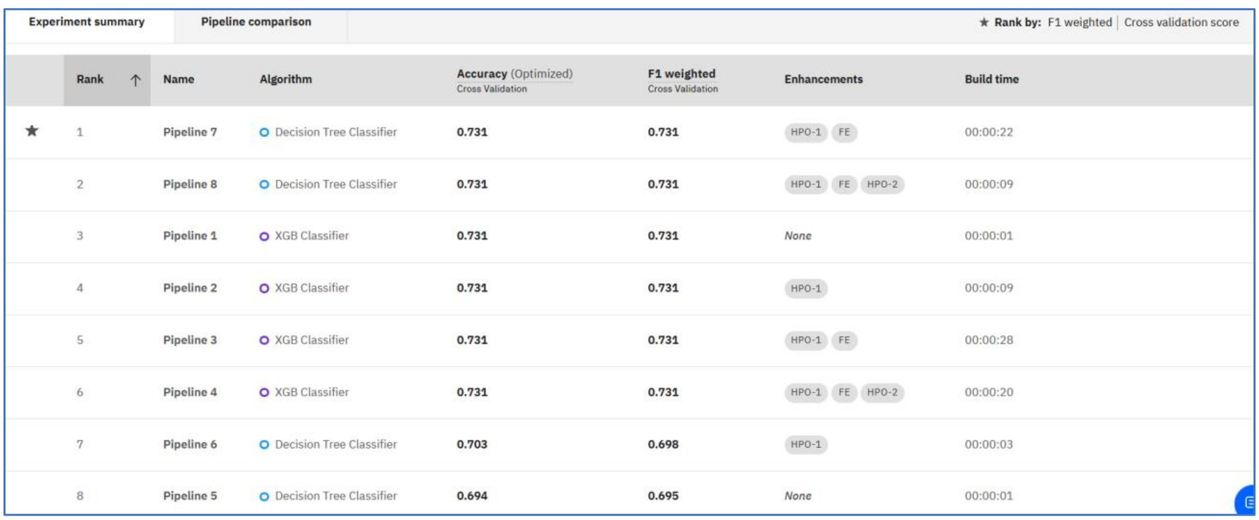
14. Scroll down to view the Pipeline Leaderboard ranked by accuracy.

	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.**

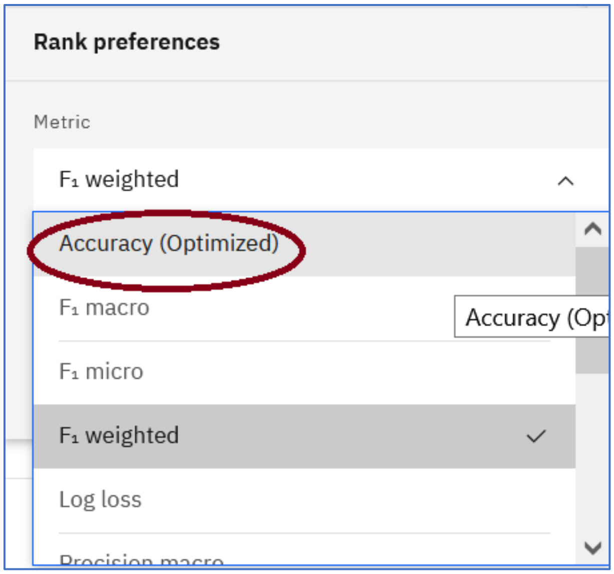


16. The list is now ranked by the F1 Weighted metric.

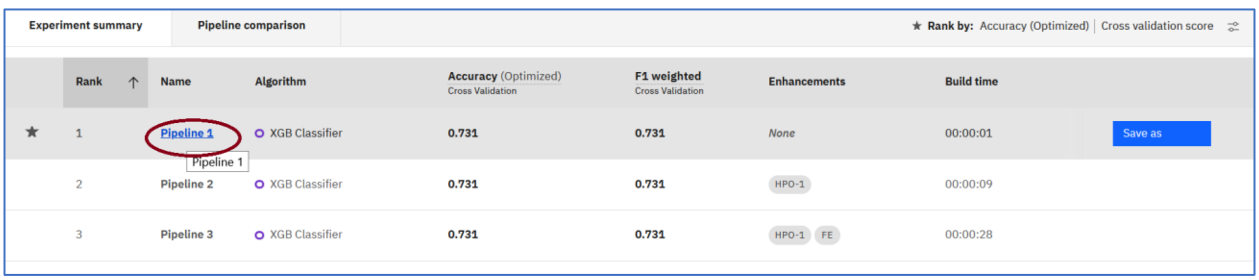


	Rank	Name	Algorithm	Accuracy (Optimized) Cross Validation	F1 weighted Cross Validation	Enhancements	Build time
★	1	Pipeline 7	Decision Tree Classifier	0.731	0.731	HPO-1 FE	00:00:22
	2	Pipeline 8	Decision Tree Classifier	0.731	0.731	HPO-1 FE HPO-2	00:00:09
	3	Pipeline 1	XGB Classifier	0.731	0.731	None	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

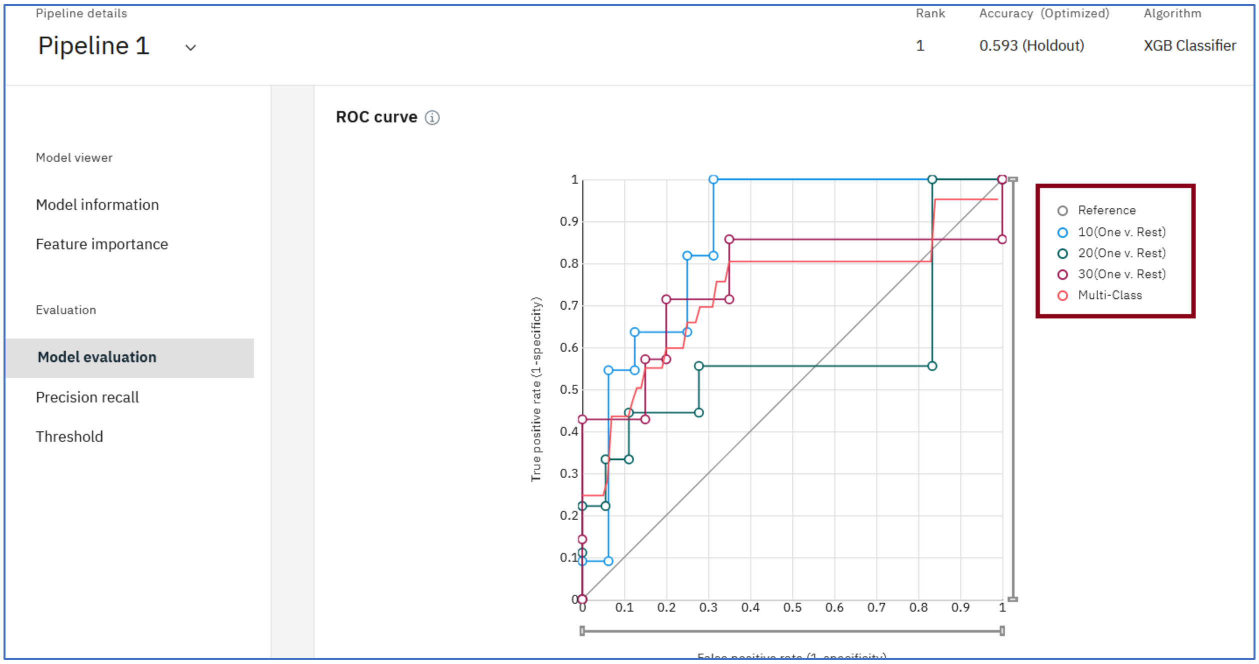
17. Return to the Accuracy ranked list by clicking on the and selecting **Accuracy**



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



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.



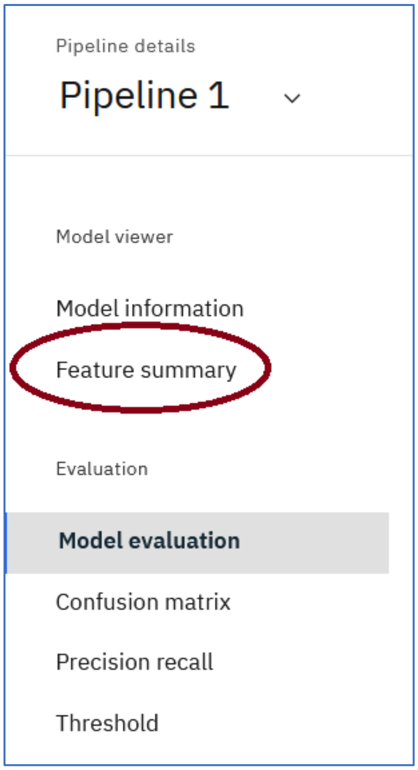
20. Scroll down to view model metrics for both the training sample and the holdout sample.

View

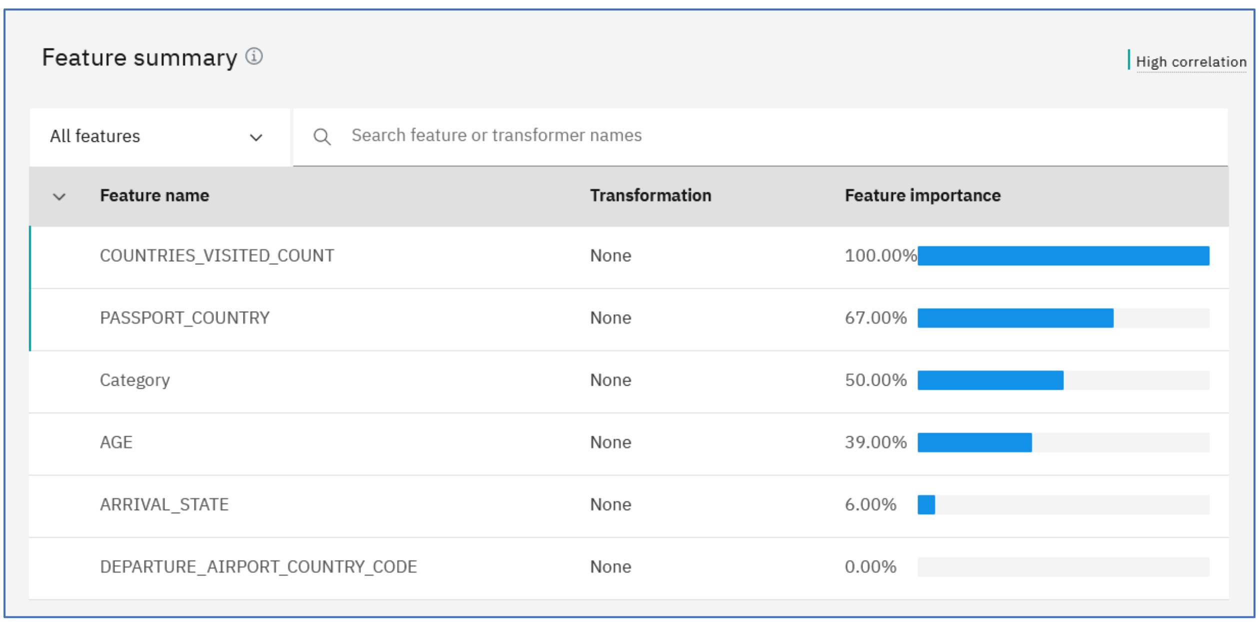
Multi-Class

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**.



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



Step 3: Save and Deploy the Model

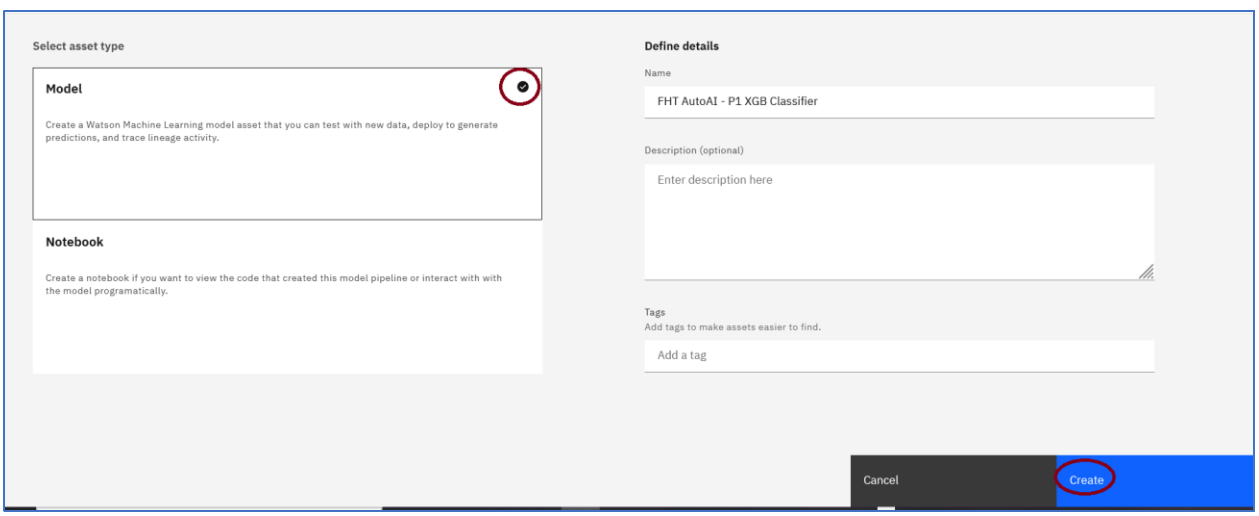
In this section, we will save and then deploy the model. You create a deployment for a machine learning model so you can submit new data to a model and get a score, or prediction back. To deploy the model, you must have a *deployment space* where you can organize the assets you need to create and monitor 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.**

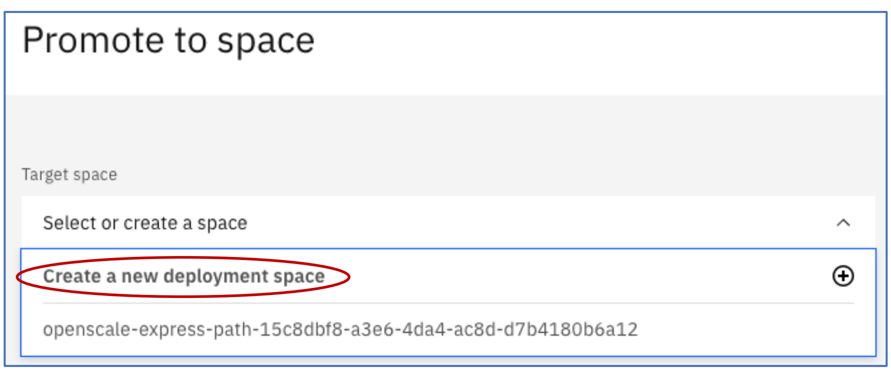
2. AutoAI can generate both a machine learning model as well as a Jupyter notebook. The Jupyter notebook can contain the source code to build the model, or the source code plus the deployment code. Feel free to experiment with saving the notebooks. For this lab, we will save the model. Make sure the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



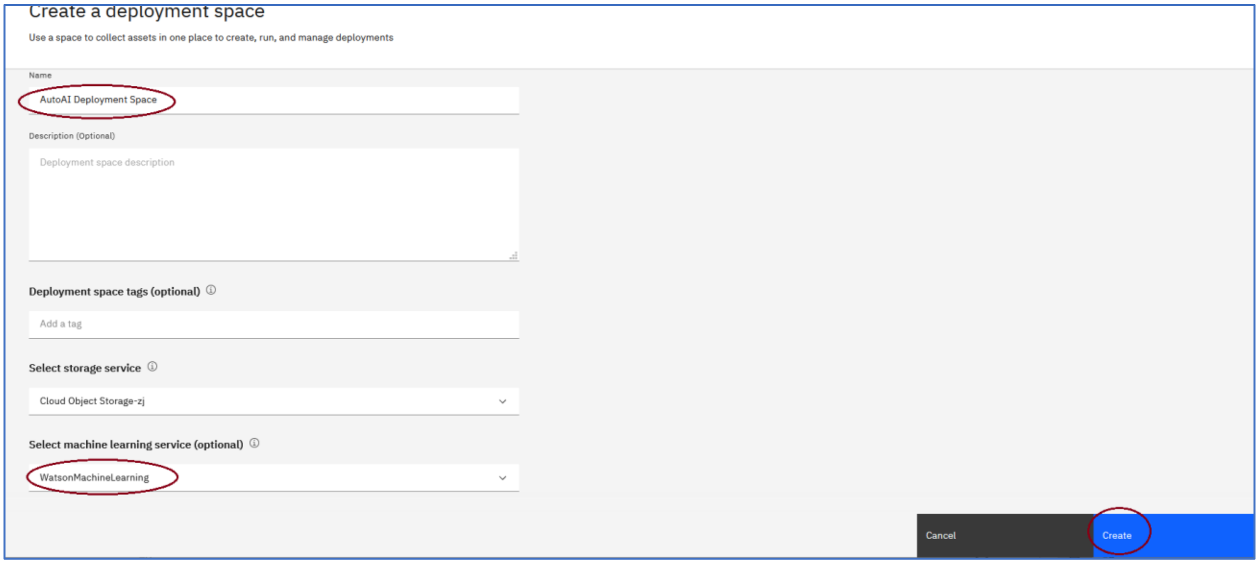
3. A message is displayed showing the model was successfully saved. Click on **View in project.**

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

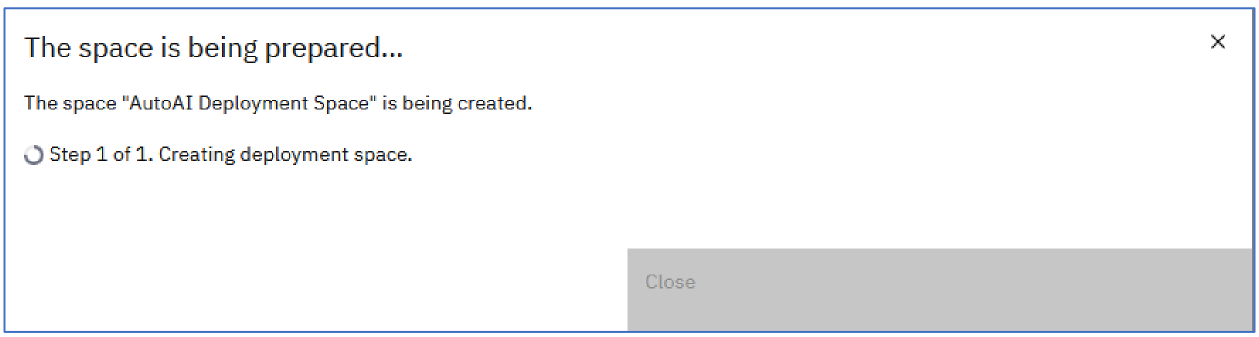
5. We will create a new space for our model deployment. Click on **New space**.

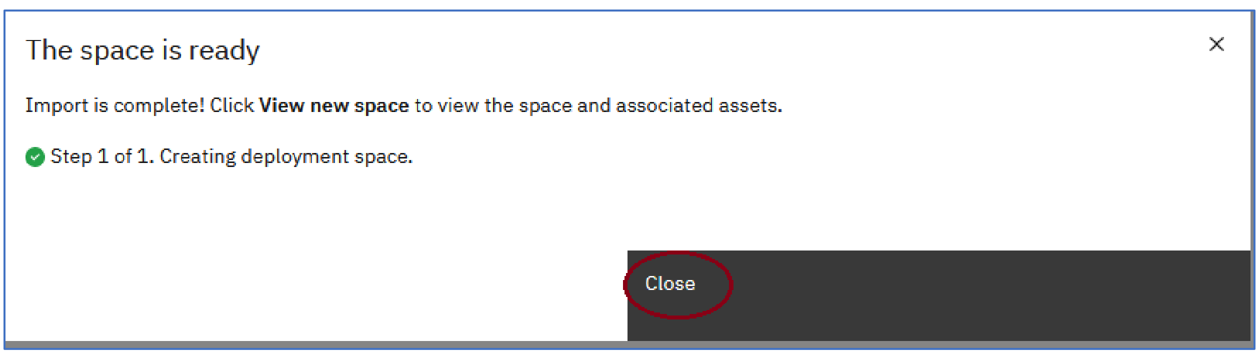


6. Enter **AutoAI Deployment Space** for the Name, scroll down and select **WatsonMachineLearning** for the **Select machine learning service**, and click **Create**.

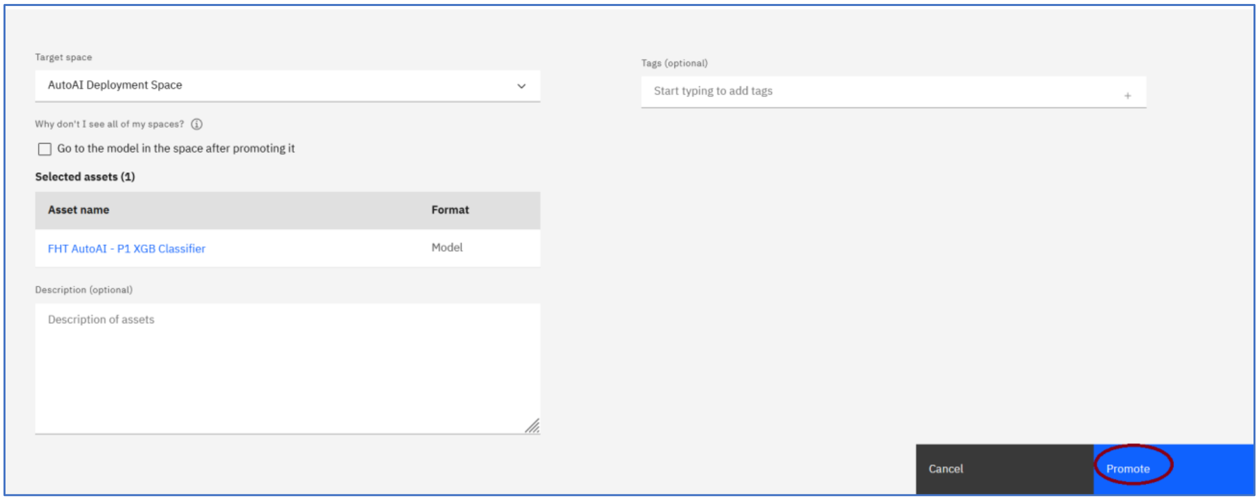


7. Two message boxes are displayed. Click **Close** in the second message box.

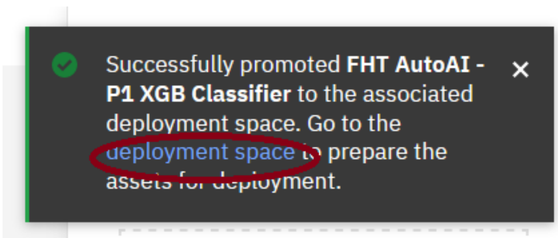




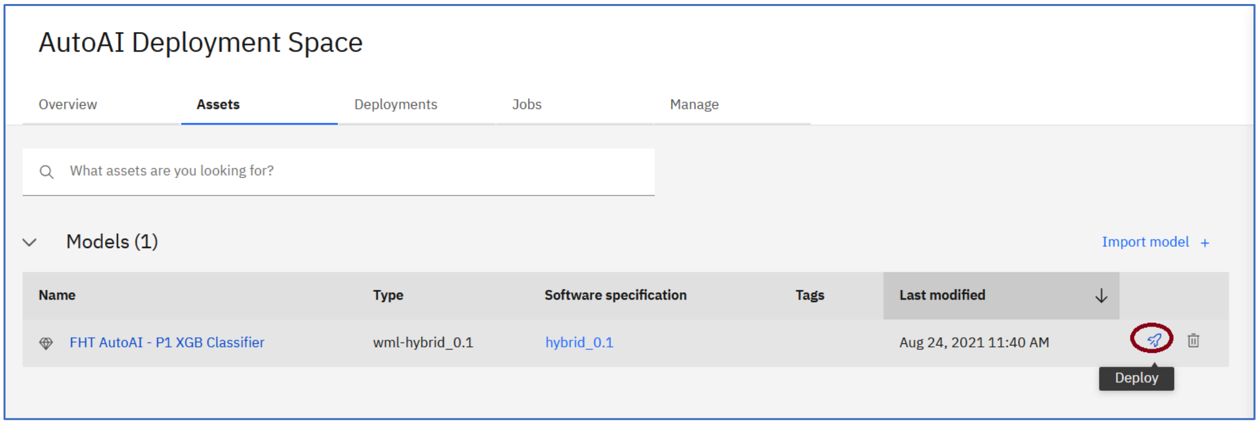
8. Click **Promote**.



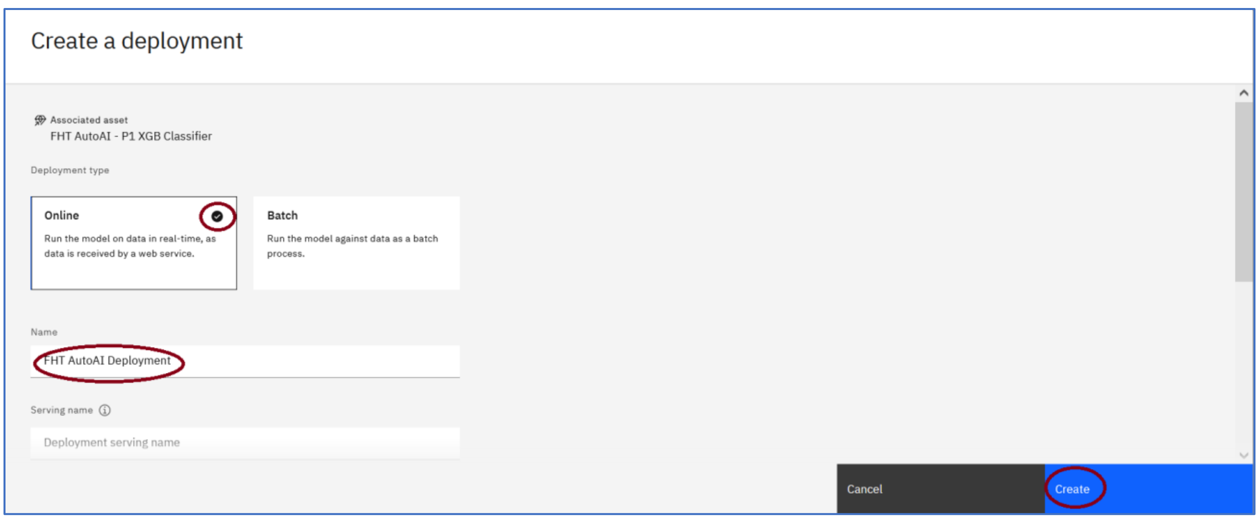
9. A message is displayed. Click **Deployment Space** in the message.



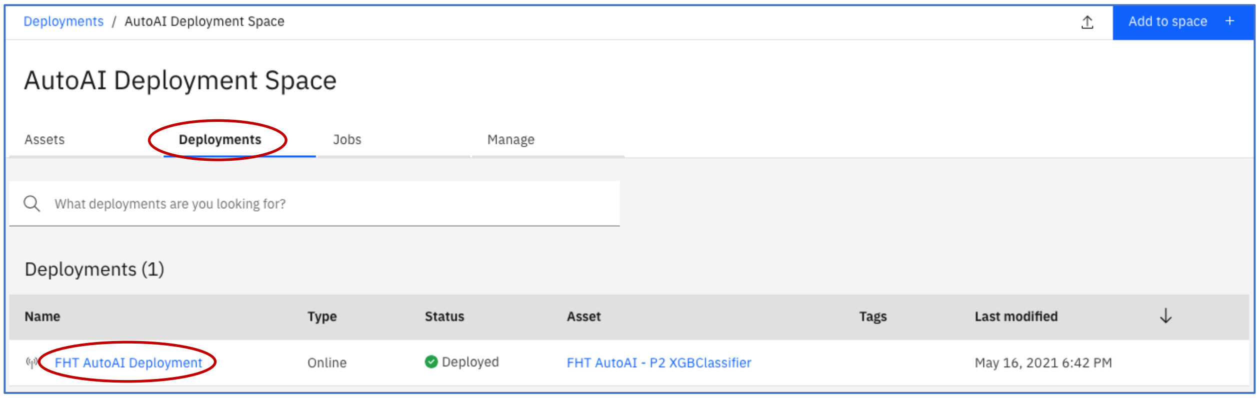
10. Hover over the promoted model until you see the deploy icon appear on the right. Click on the icon.



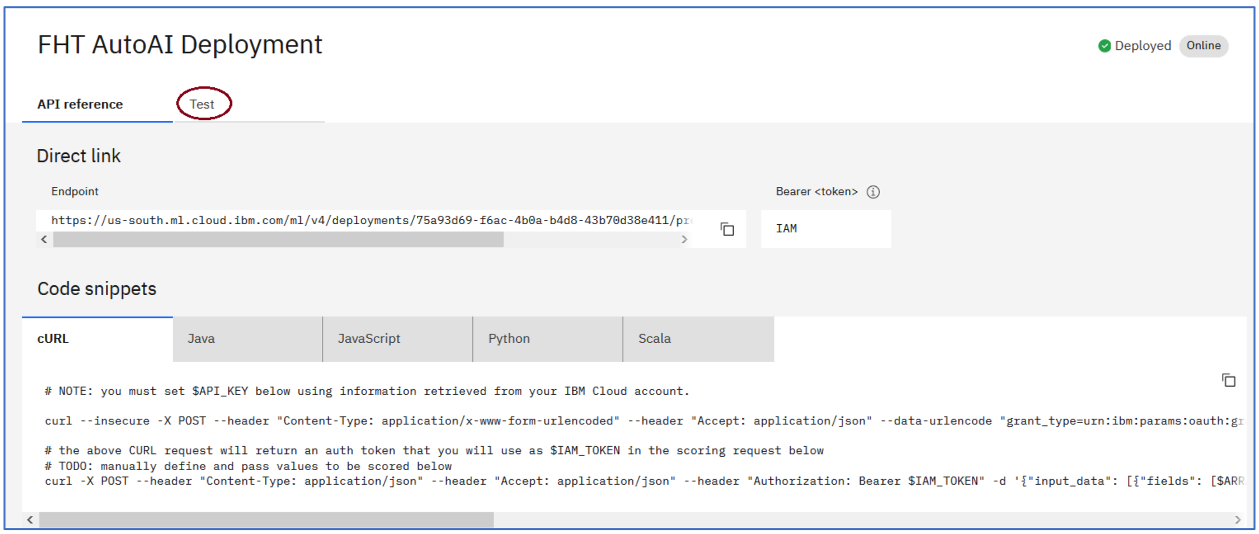
11. Click **Online** for the **Deployment type**, enter **FHT AutoAI Deployment** for the **Name**, and click **Create**.



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.



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.



14. Enter the following values:

PASSPORT_COUNTRY – **Ghana**

COUNTRIES_VISITED_COUNT – **3**

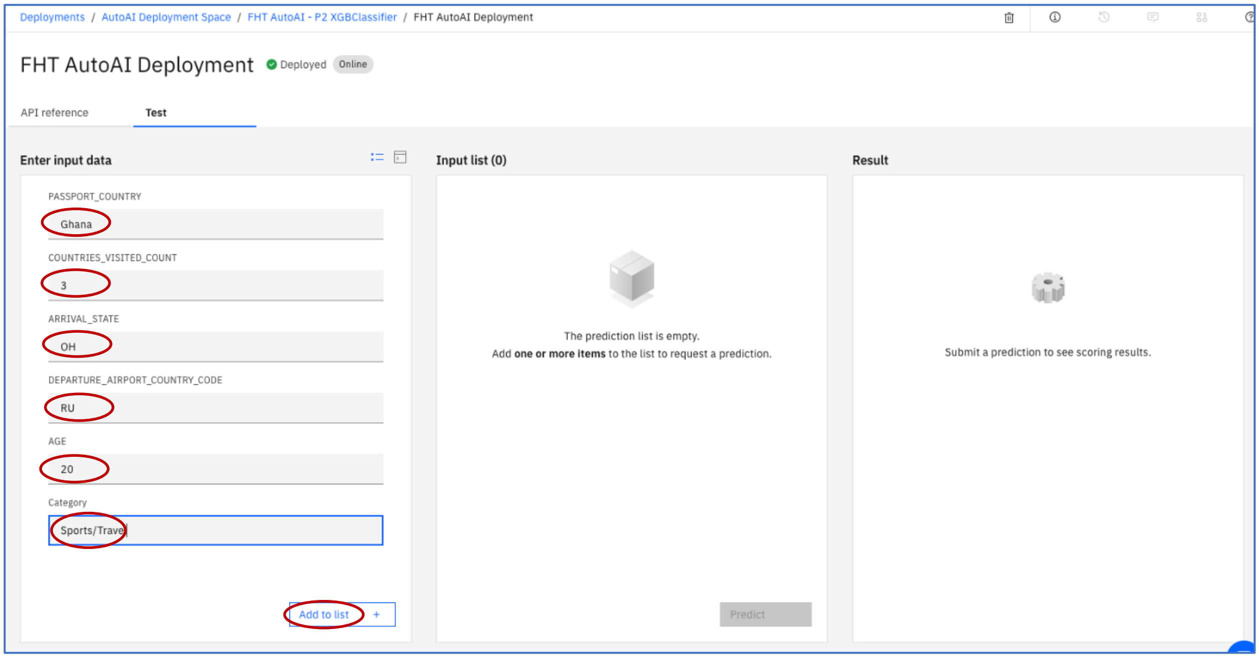
ARRIVAL_STATE – **OH**

DEPARTURE_AIRPORT_COUNTRY_CODE – **RU**

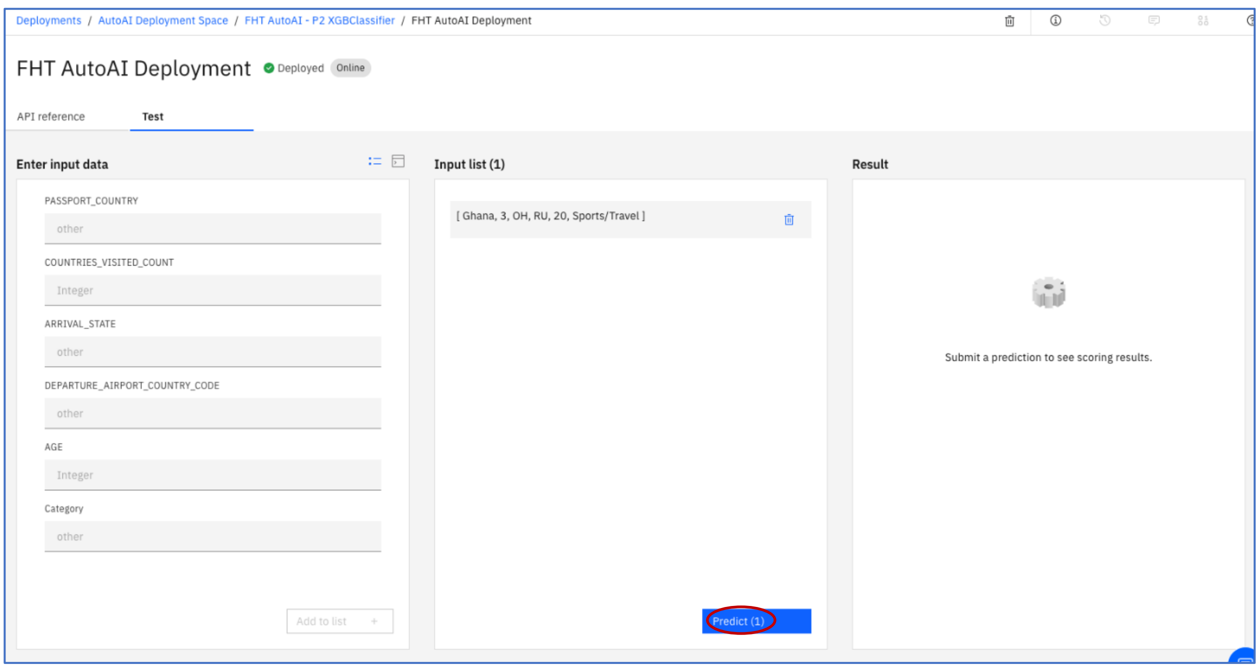
AGE – **20**

Category – **Sports/Travel**

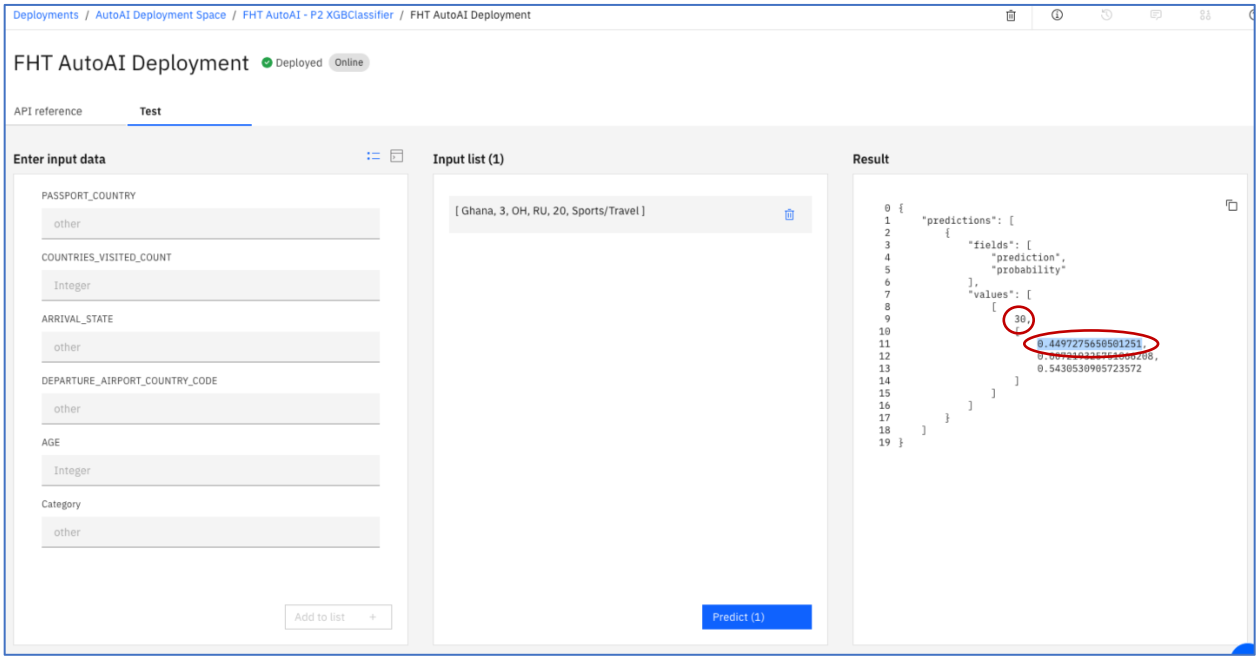
Click **Add to list**.



15. Click Predict (1)



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



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