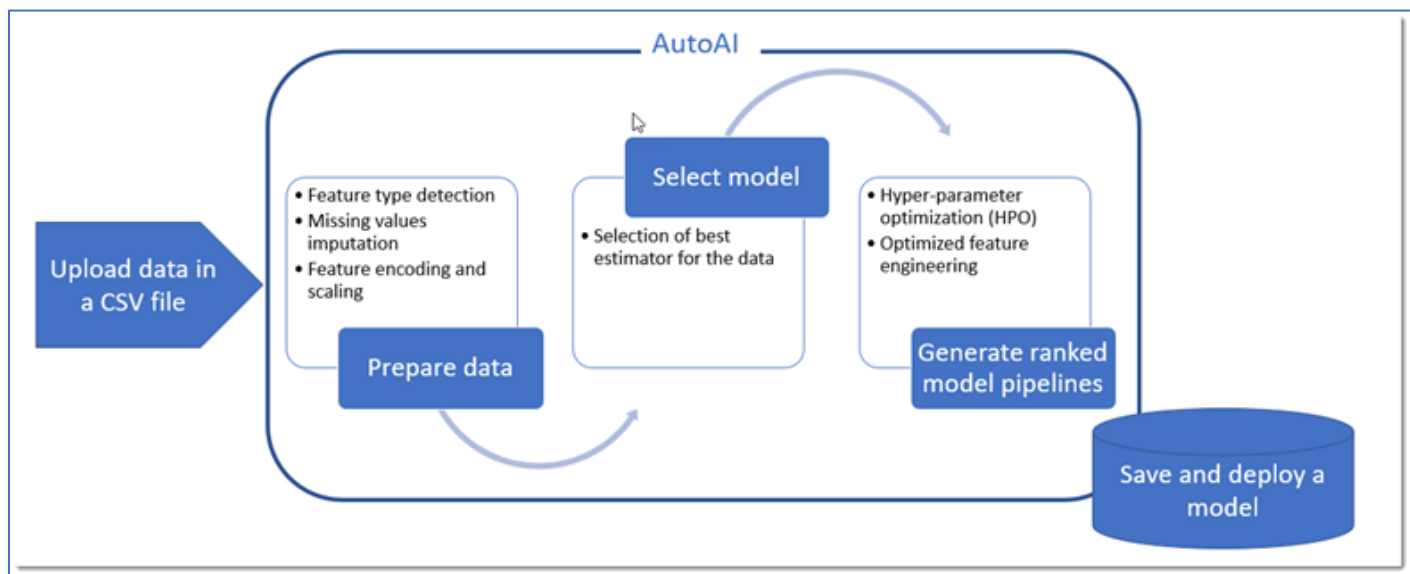


# AutoAI Lab

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

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 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** - 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.

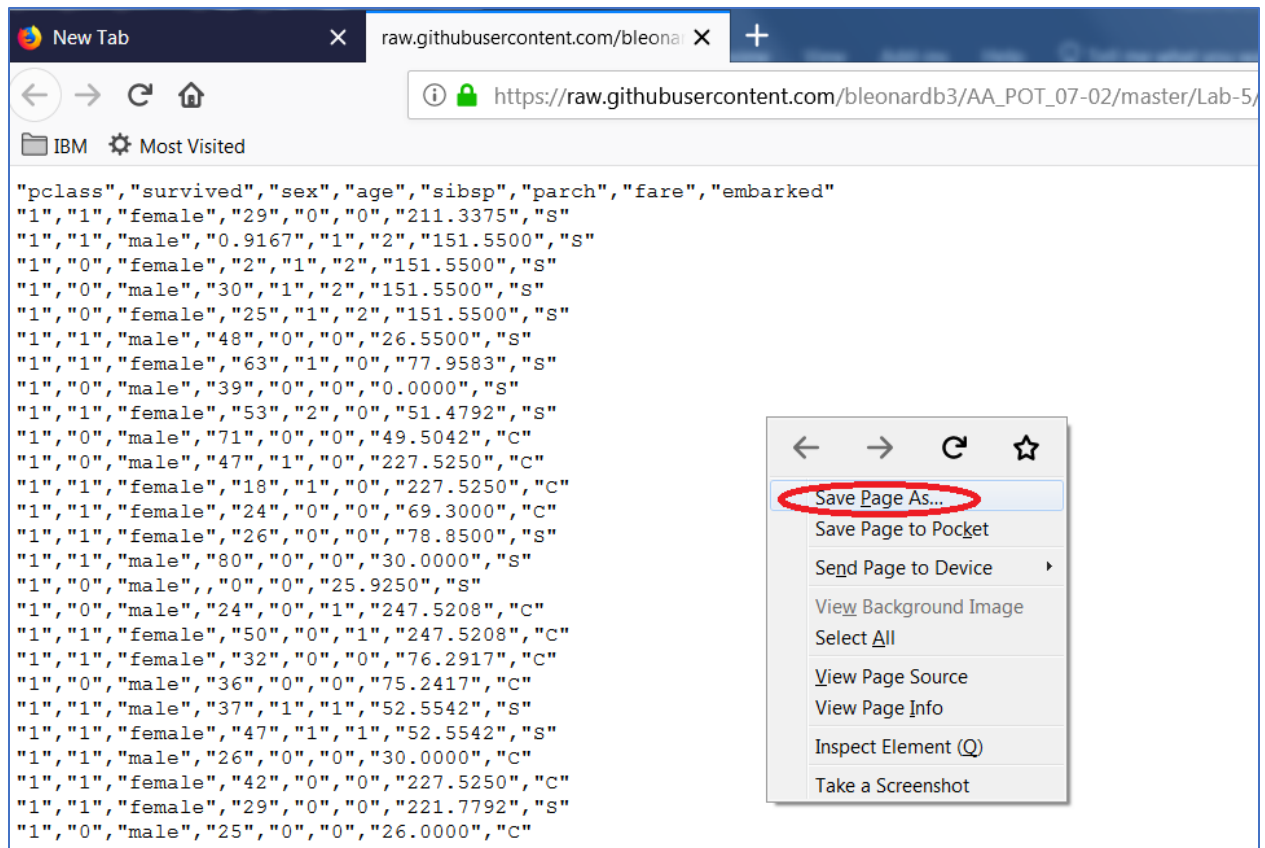
- **Hyperparameter optimization** - Hyper-parameter optimization refines the best performing model pipelines. AutoAI uses a novel hyper-parameter optimization algorithm optimized for costly function evaluations such as model training and scoring that are typical in machine learning. This approach enables 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:

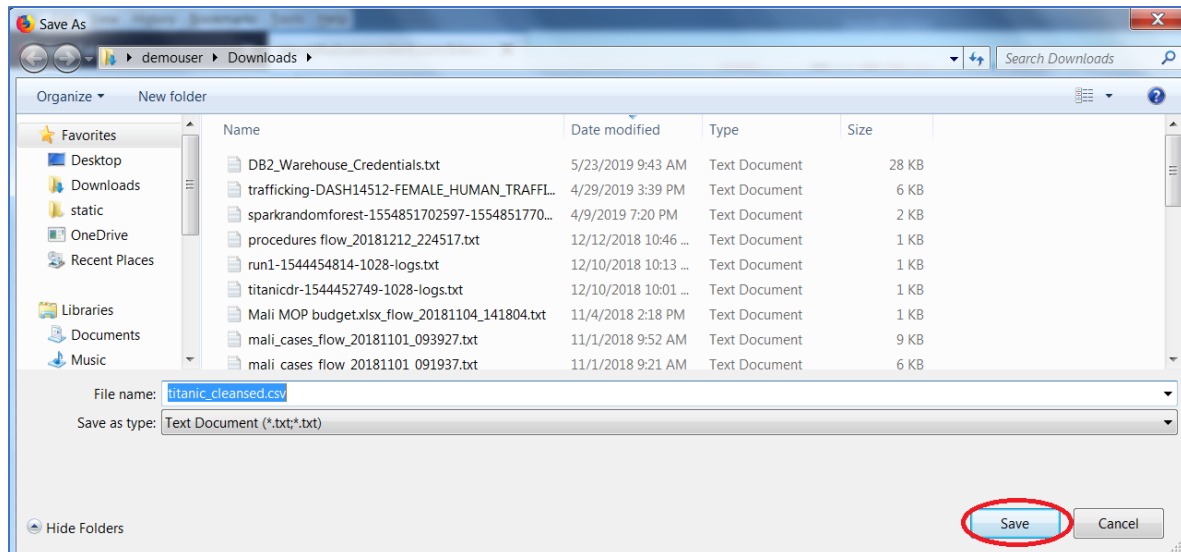
1. Download a Titanic cleansed data set
2. Add an Auto AI Experiment
3. Save and Deploy the selected model
4. Test the Deployment


### Step 1: Download the `titanic_cleansed.csv` data set

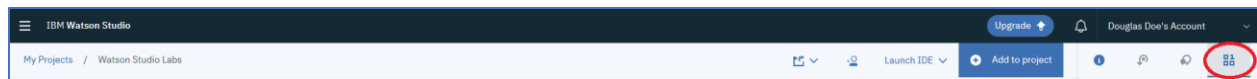
1. Download the **titanic\_cleansed.csv** data file from the following location by clicking on the link [here](#). Note this is a different file than used in the previous labs.
2. Right-click on the window, and click **Save Page As...**



3. Click on **Save**. Note, if the file is named `titanic_cleansed.csv.txt`, change it to be `titanic_cleansed.csv`.

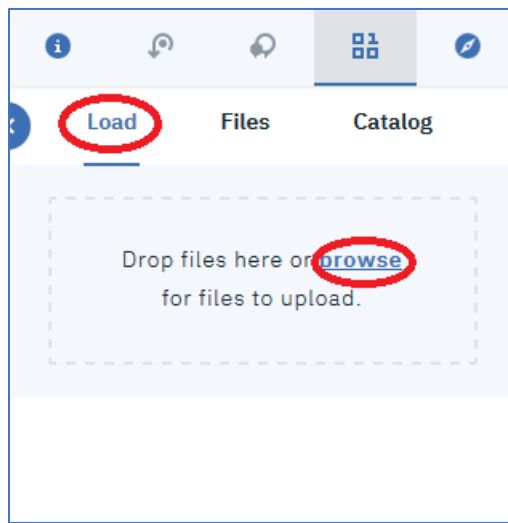


4. Go back to your Watson Studio Labs project. Click on the  icon.

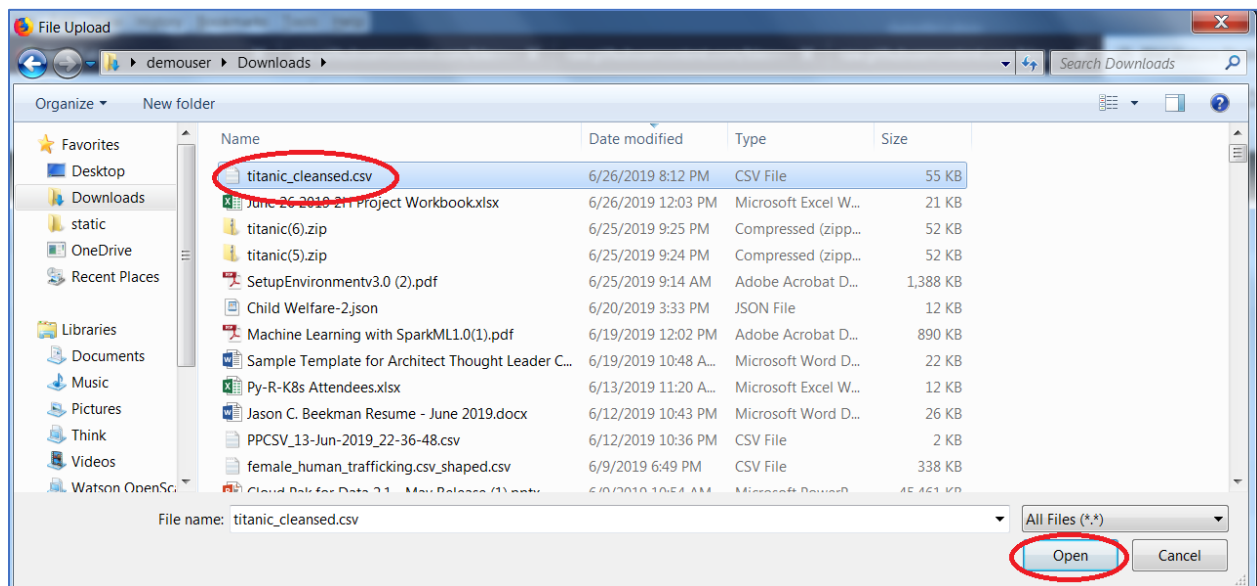


- Click on the **Load** tab and then click on **browse**. If you don't see the **Load** tab, click on

the  icon again.



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



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

My Projects / Watson Studio Labs

Overview Assets Environments Bookmarks Deployments Access Control Settings

What assets are you looking for?

▼ Data assets

0 asset selected.

<input type="checkbox"/>	NAME	TYPE	CREATED BY	LAST MODIFIED	ACTIONS
<input type="checkbox"/>	CSV titanic_cleansed.csv	Data Asset	Douglas Doe	26 Jun 2019, 12:35:27 pm	
<input type="checkbox"/>	CSV 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

Overview Assets Environments Bookmarks Deployments Access Control Settings

Launch IDE ▼ Add to project

2. Click on **AutoAI experiment**

Choose asset type

Available asset types

Data	Connection	Connected data
<b>AutoAI experiment</b>	Notebook	Dashboard
Visual Recognition m...	MDM Configuration	Natural Language Cla...
Model from file	DataStage flow	Federated Learning e...
Deep learning experi...	Modeler flow	Metadata Import
Data Refinery flow	Decision Optimizatio...	

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

**Define details**

Name \*  
Titanic AutoAI

Description  
Description of AutoAI experiment

**Associate 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.

Cancel Create

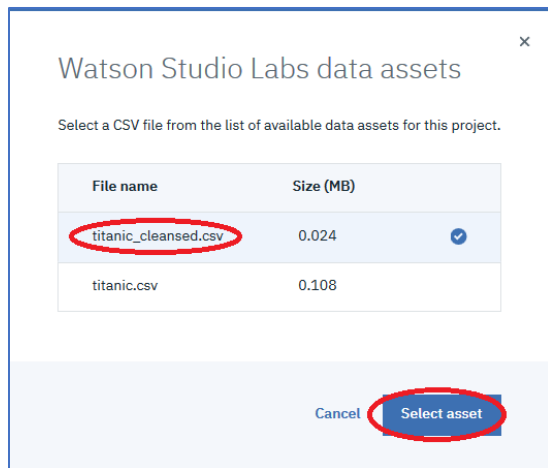
4. Click on **Select from project**.

**Add data source**

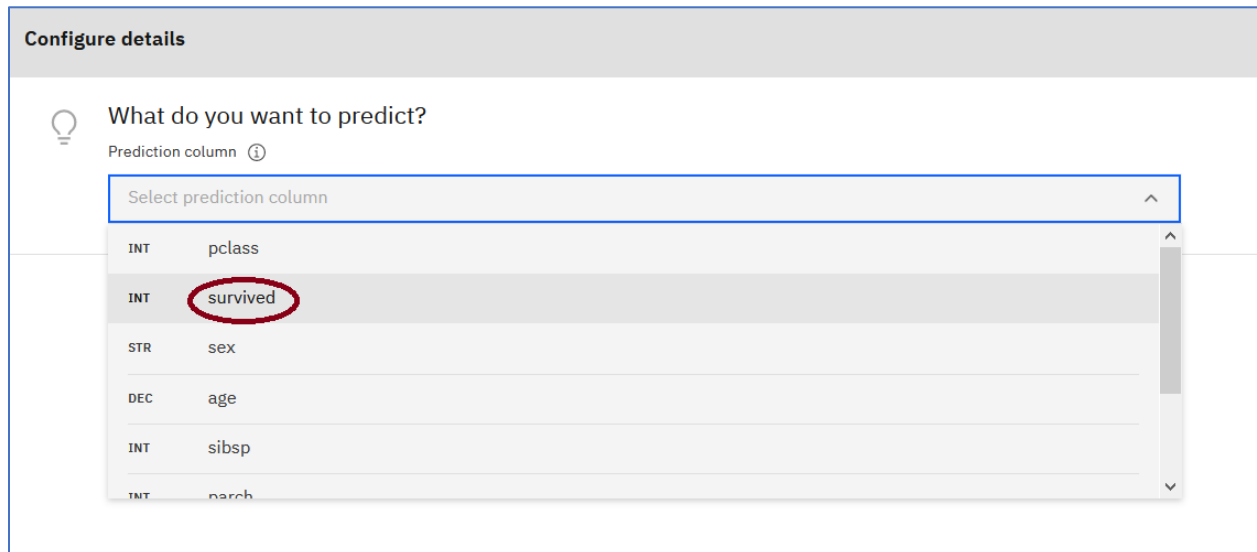
Drop or browse for a csv file.

Browse or Select from project

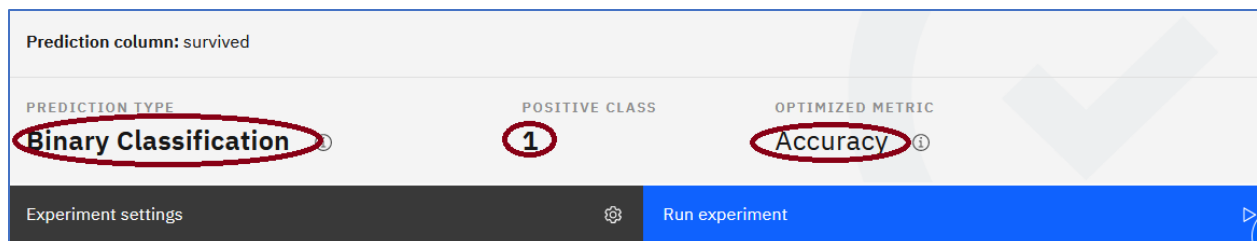
5. Click on **titanic\_cleansed.csv** and then click on **Select asset**.



6. Click on **survived** as the column to predict.



7. Note, based on this selection, the **Prediction Type** is **Binary Classification**, and the **Optimized Metric** is **Accuracy**. Further note, the **Positive Class** is correctly defaulted as “1” – survived.



8. Click on **Experiment settings** to change the default optimized metric.

PREDICTION TYPE	POSITIVE CLASS	OPTIMIZED METRIC
Binary Classification ⓘ	1	Accuracy ⓘ
Experiment settings ⓘ	⚙️	Run experiment ▶

9. Click on **Prediction**.

Experiment settings

Data source

**Prediction**

Runtime

10. Click on **ROC AUC** (Receiver Operating Characteristic Area Under the Curve) and then click on **Save Settings**

Experiment settings	1
Data source	
<b>Prediction</b>	
Runtime	

**Optimized metric**  
Choose the metric to optimize for the experiment.

Accuracy (Recommended))	
<b>ROC AUC</b>	✓
Average Precision	
Precision	
Recall	
F1	
Log Loss	

**Algorithms to test**  
Select which of the following algorithms is to be considered when the experiment is run. The list of algorithms are based on the selected prediction type.

Included algorithms 7/7

🔍 Search by algorithm

Cancel

**Save settings**

11. Click on **Run experiment**.



Configure details

💡

What do you want to predict?

Prediction column ⓘ

survived

✕

▼

Prediction column: survived

PREDICTION TYPE

Binary Classification ⓘ

POSITIVE CLASS

1

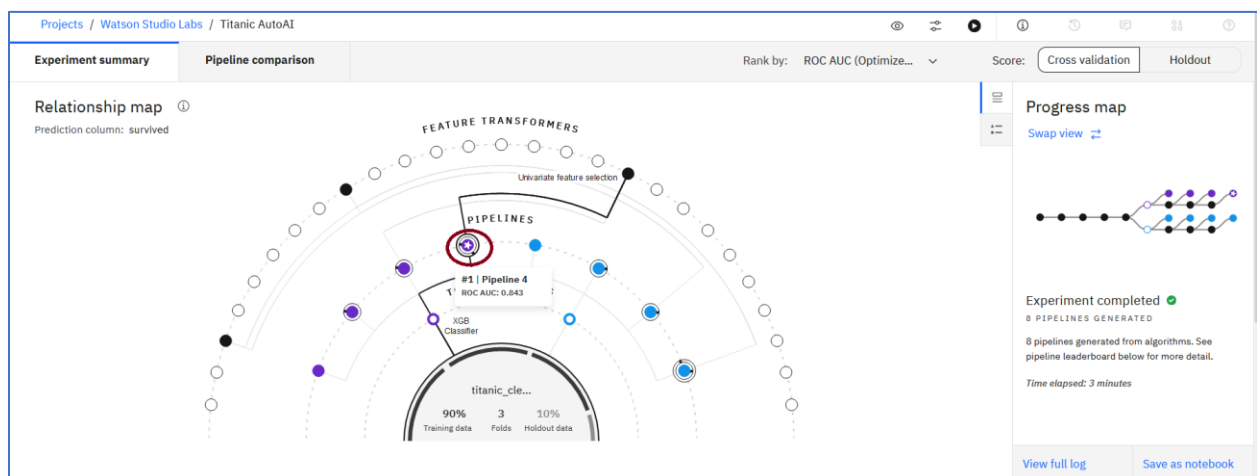
OPTIMIZED METRIC

ROC AUC ⓘ

Experiment settings ⓘ

Run experiment

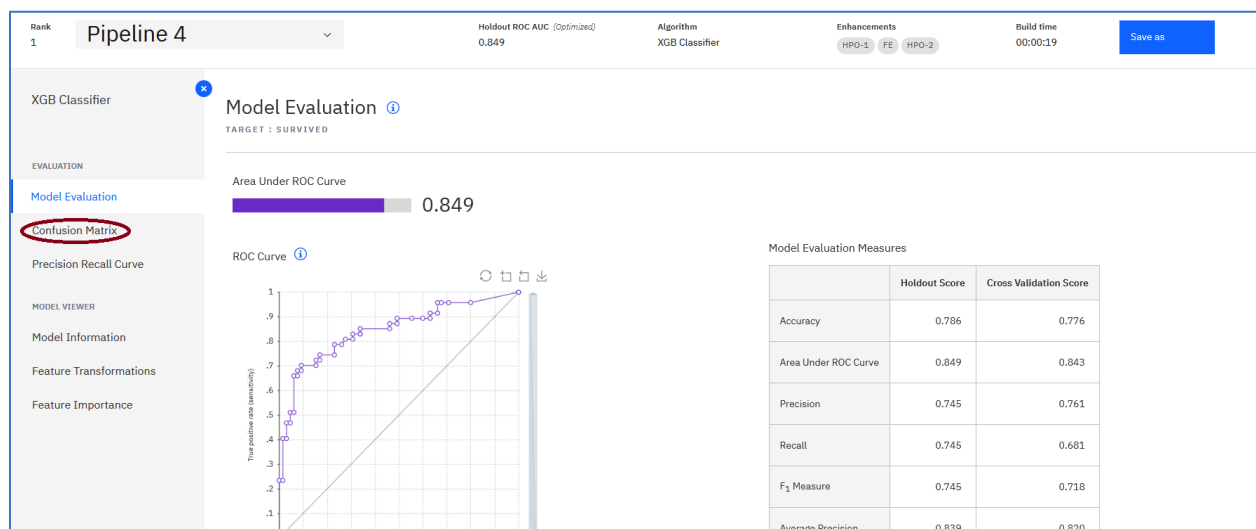
12. It will take several minutes for the eight alternative pipelines to be analyzed. The first pipeline picks the best algorithm. The second pipeline optimizes the hyper-parameters for the selected algorithm. The third pipeline does a feature transformation to try to improve the performance of the algorithm. The fourth pipeline repeats the hyper-parameter tuning with the new set of features. The next 4 pipelines do the same thing for the second best algorithm. Note, you can move ahead after 4 of the 8 pipelines have been completed.



13. Scroll down to view the **Pipeline leaderboard**. The pipeline summary is then displayed. Click on the right arrow **Pipeline 4**.

Rank	↑	Name	Algorithm	ROC AUC (Optimized)	Enhancements	Build time
★ 1		Pipeline 4	XGB Classifier	0.843	HPO-1 FE HPO-2	00:00:19
2		Pipeline 3	XGB Classifier	0.842	HPO-1 FE	00:00:40
3		Pipeline 2	XGB Classifier	0.840	HPO-1	00:00:09
4		Pipeline 1	XGB Classifier	0.836	None	00:00:01
5		Pipeline 8	Logistic Regression	0.835	HPO-1 FE HPO-2	00:00:09
6		Pipeline 7	Logistic Regression	0.835	HPO-1 FE	00:00:17
7		Pipeline 5	Logistic Regression	0.832	None	00:00:01
8		Pipeline 6	Logistic Regression	0.832	HPO-1	00:00:03

14. Metrics are displayed for both the holdout sample and the training sample (cross-validation). Click on **Confusion Matrix**.

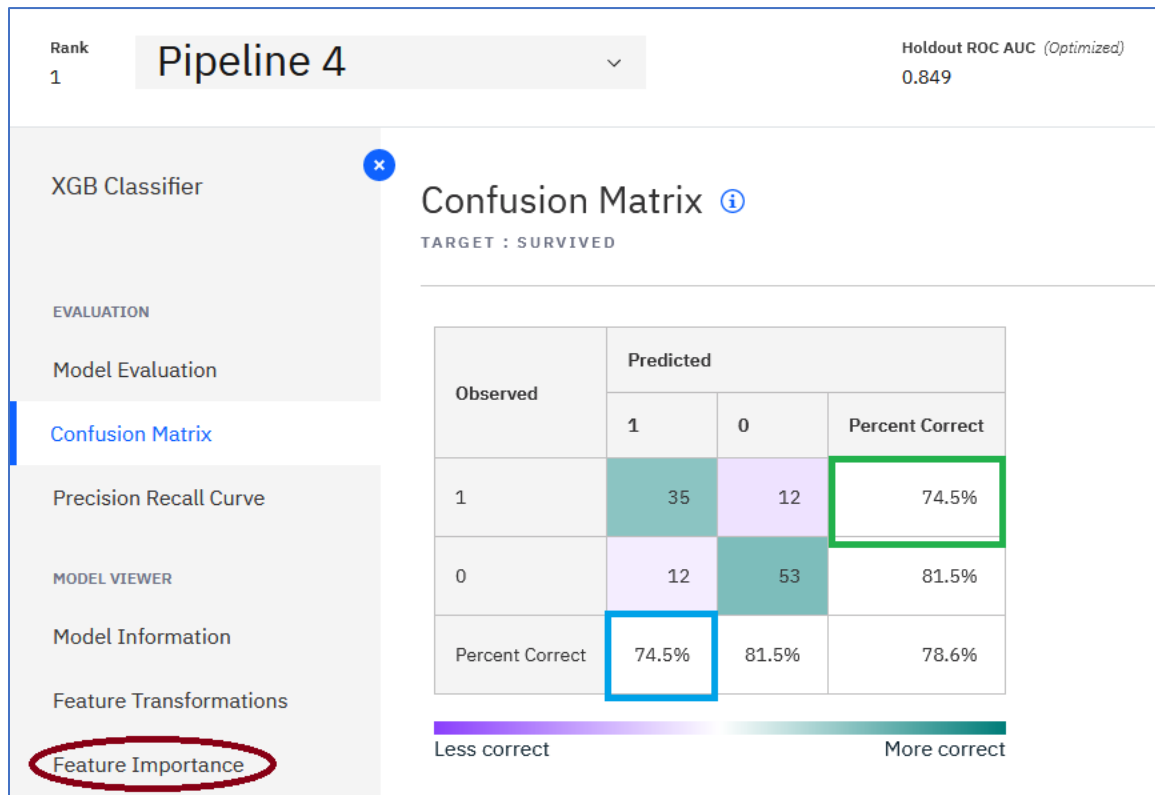


15. The Confusion Matrix is displayed for the holdout sample. The different metrics are computed based on the numbers in the Confusion Matrix. For example, Precision is defined by the percentage of predicted positives that are actually positive (i.e. the percentage of predicted survivors that survived). 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.

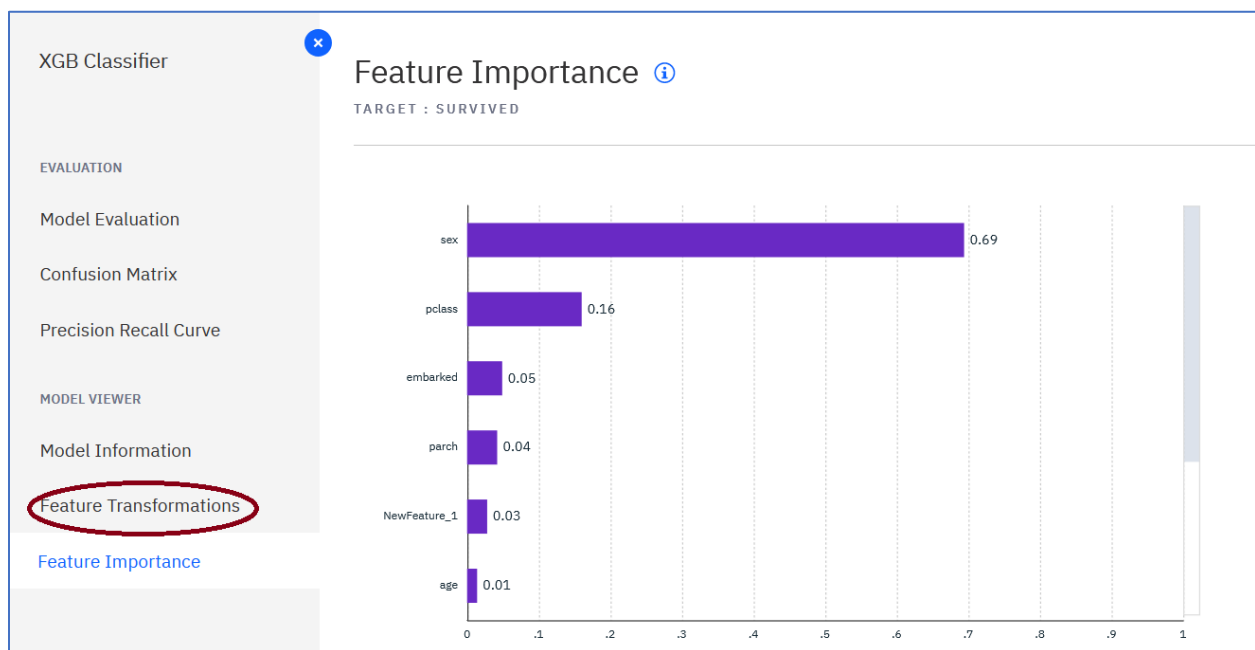
Precision = True Positive / (True Positive + False Positive) – shown inside blue rectangle on diagram below.

Recall = True Positive / (True Positive + False Negative) - shown inside green rectangle on diagram below. This is also called the True Positive Rate.

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



16. According to the Feature Importance, the sex variable is considered the most important feature followed by the passenger class. After reviewing the Feature Importance, click on **Feature Transformations**



17. Two new features are derived as shown below.

Feature Transformations ⓘ		
TARGET : SURVIVED		
New Feature	Original Feature	Transformation
NewFeature_1	isoforestanomaly_0-3	isoforestanomaly_0-3
NewFeature_0	isoforestanomaly_0	isoforestanomaly_0

### Step 3 – Save and Deploy the Selected Model

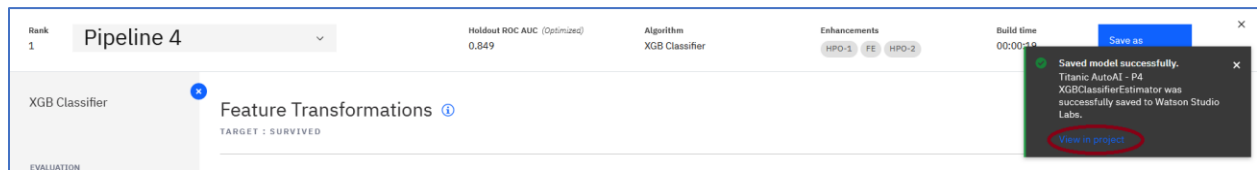
1. Click on **Save as**.

Rank 1	Pipeline 4	Holdout ROC AUC (Optimized) 0.849	Algorithm XGB Classifier	Enhancements HPO-1 FE HPO-2	Build time 00:00:19	<b>Save as</b>
-----------	------------	--------------------------------------	-----------------------------	--------------------------------	------------------------	----------------

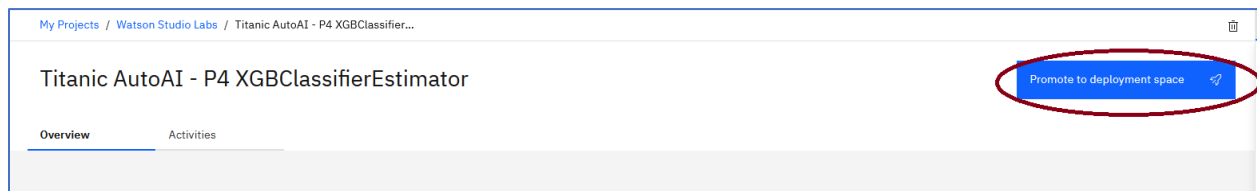
2. You have the option to save a Model or a Notebook. The notebook contains the code used to generate the pipeline. In this way a data scientist could use this as a starting point to tune the model even further. We will save the model. Optionally change the default name and click on **Save**.

<b>Select asset type</b>	<b>Define details</b>
<b>Model</b> Create a Watson Machine Learning model asset that you can test with new data, deploy to generate predictions, and trace lineage activity.	Name Titanic AutoAI - P4 XGBClassifierEstimator
<b>Notebook</b> beta Create a notebook if you want to view the code that created this model pipeline or interact with the model programmatically.	Description (optional) Enter description here
	Tags Add tags to make assets easier to find. Add a tag
	Cancel Create

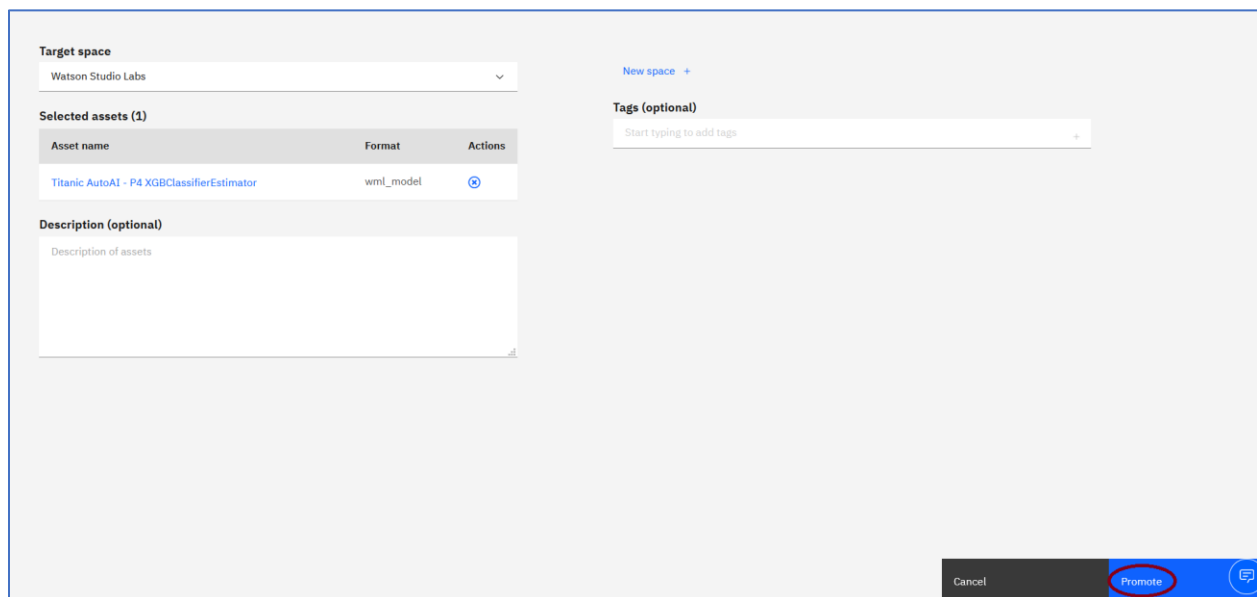
3. The model is successfully saved. Click on **View in Project**.



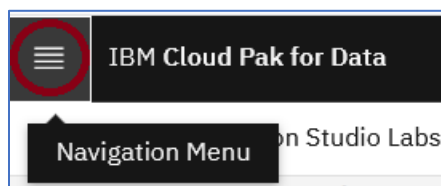
4. In order to deploy the model, we need to promote it to the deployment space that was created in Lab-1. Click on **Promote to deployment space**.



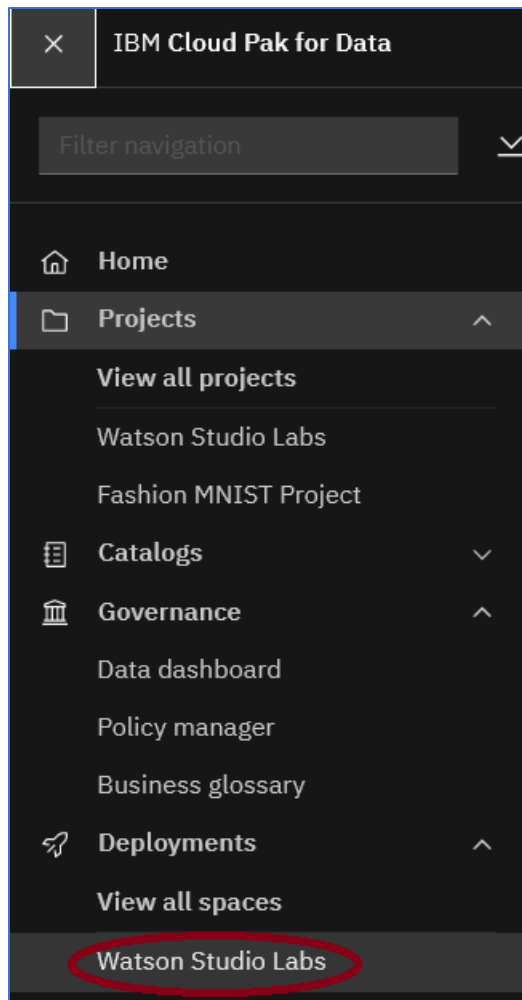
5. Click on **Promote**.




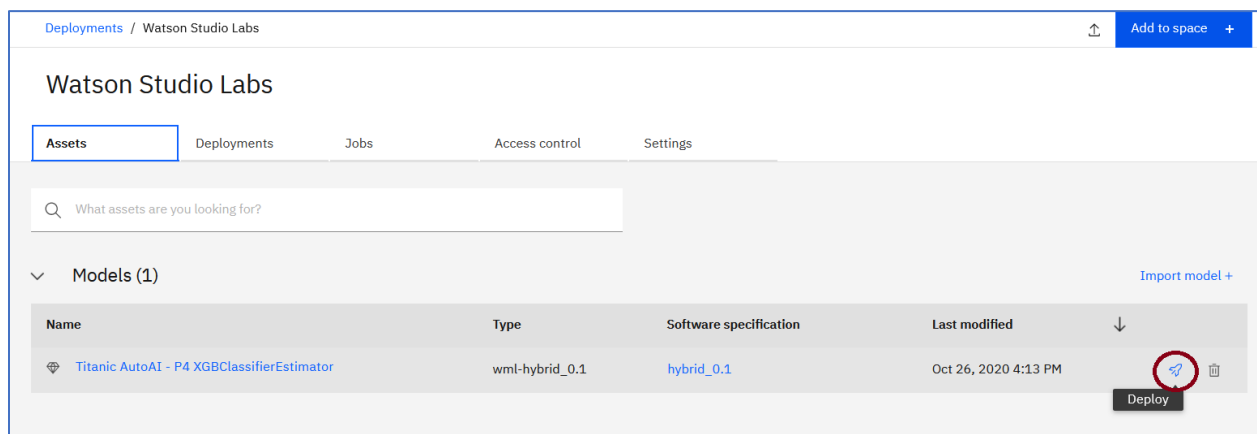
6. Click on the hamburger icon .



7. Click on **Deployments** and **Watson Studio Labs**



8. Hover over the model and on the right the Deploy icon  will appear. Click on the icon.



9. Click on **Online**, enter **Titanic AutoAI Deploy** for the **Name** of the deployment, optionally enter a **Description**, and click **Create**.

The screenshot shows the deployment configuration form for the asset "Titanic AutoAI - P4 XGBClassifierEstimator". The "Deployment type" section has two options: "Online" (selected and circled in red) and "Batch". The "Name" field contains "Titanic AutoAI Deploy" (circled in red). The "Description" field is empty. The "Software specification" is set to "hybrid\_0.1". At the bottom right, there are "Cancel" and "Create" buttons, with the "Create" button circled in red.

10. Click on the **Deployment** Tab to see the deployment status.

The screenshot shows the "Deployments" tab in Watson Studio Labs. The "Deployments (1)" section contains a table with the following data:

Name	Type	Status	Asset	Last modified
Titanic AutoAI Deploy	Online	Deployed	Titanic AutoAI - P4 XGBClassifierEstimator	Oct 26, 2020 7:01 PM

11. When the status shows **Deployed**, click on **Titanic AutoAI Deploy**

This screenshot is identical to the previous one, but the "Titanic AutoAI Deploy" link in the table is circled in red to indicate it should be clicked.

12. 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 on **Test**.





## 14. Click **Predict**

Titanic AutoAI Deploy Deployed Online

API reference **Test**

Enter input data

pclass  
Integer

sex  
other

age  
Double

sibsp  
Integer

parch  
Integer

fare  
Double

Add to list +

Input list (1)

[ 1, female, 5, 1, 2, 23, S ]

Predict (1)

15. The model predicts this passenger would survive with 98% confidence.

Titanic AutoAI Deploy Deployed Online

API reference **Test**

Enter input data

pclass  
Integer

sex  
other

age  
Double

sibsp  
Integer

parch  
Integer

fare  
Double

Add to list +

Input list (1)

[ 1, female, 5, 1, 2, 23, S ]

Predict (1)

Result

```
0 {
1   "predictions": [
2     {
3       "fields": [
4         "prediction",
5         "probability"
6       ],
7       "values": [
8         [
9           1,
10          [
11            0.01926928667086070,
12            0.9807307133291393
13          ]
14        ]
15      ]
16    }
17  ]
18 }
```

**You have successfully completed the lab!!!**

- ✓ Downloaded a Titanic cleansed data set
- ✓ Added an Auto AI Experiment
- ✓ Saved and Deployed the selected model.
- ✓ Tested the Deployment.

