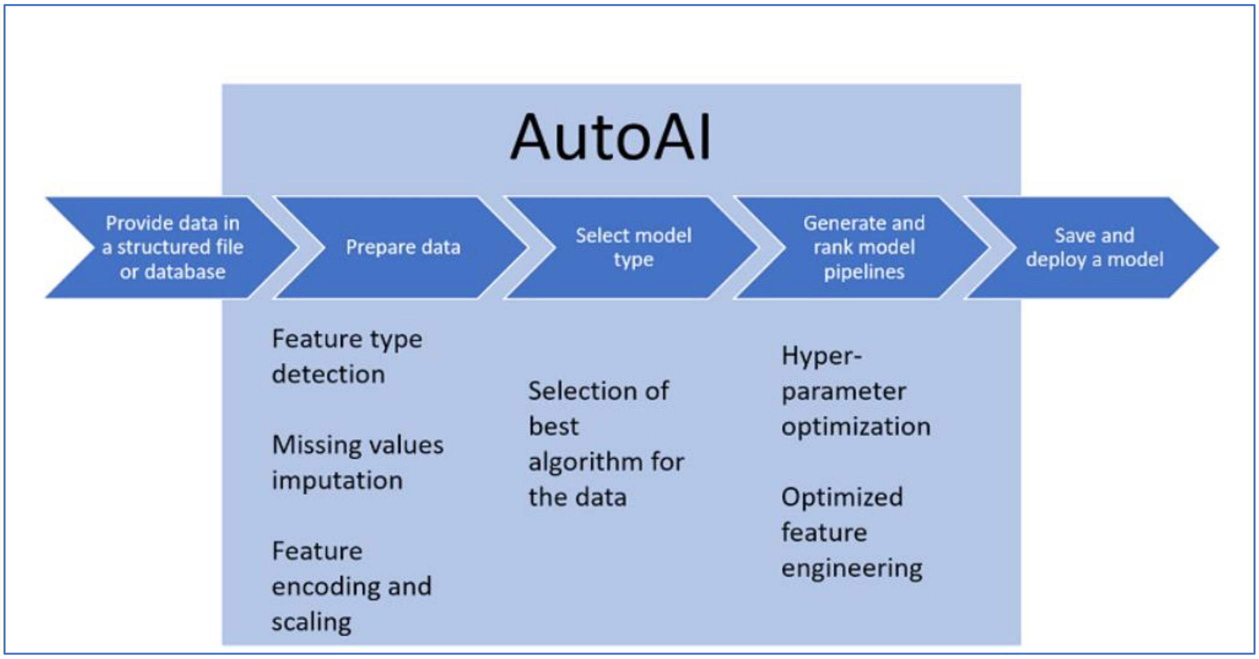
AutoAI Lab

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.

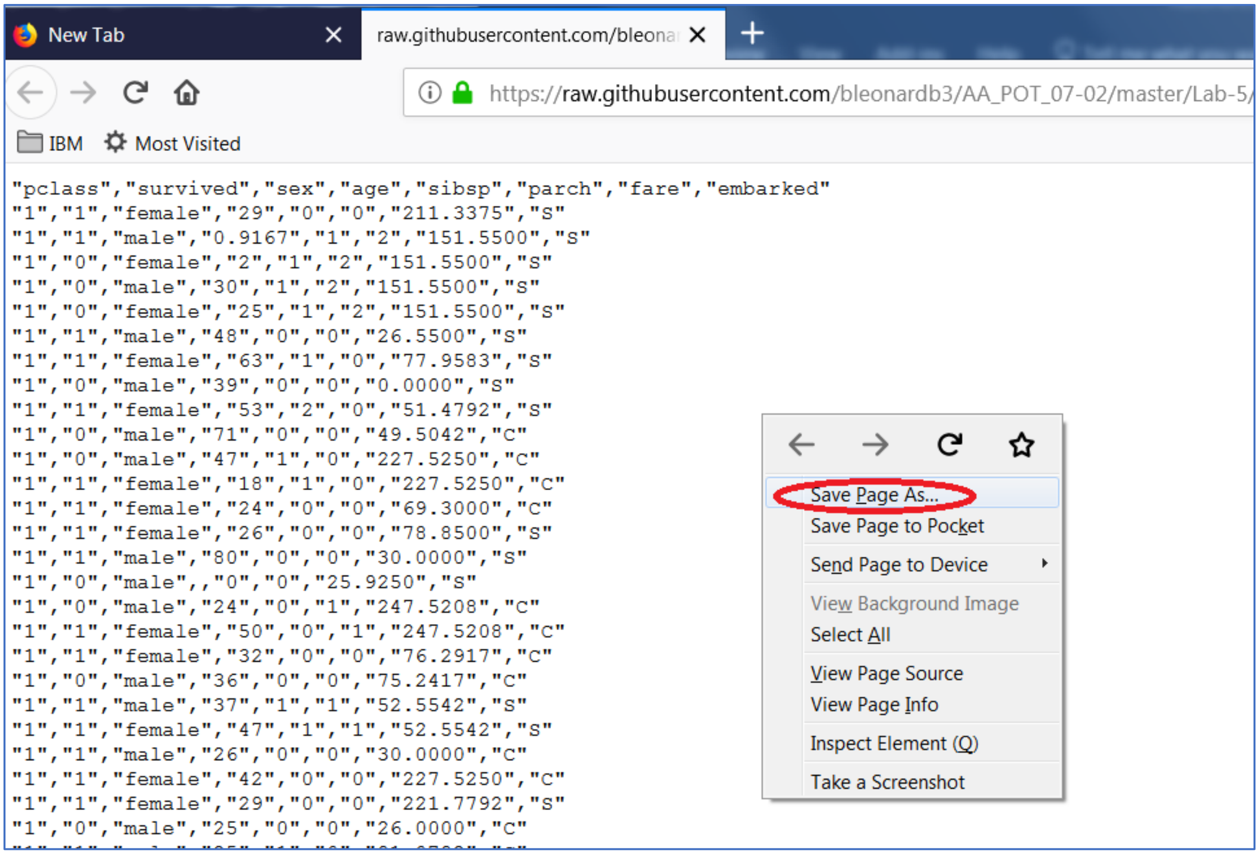
We will perform the following steps in this lab:

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

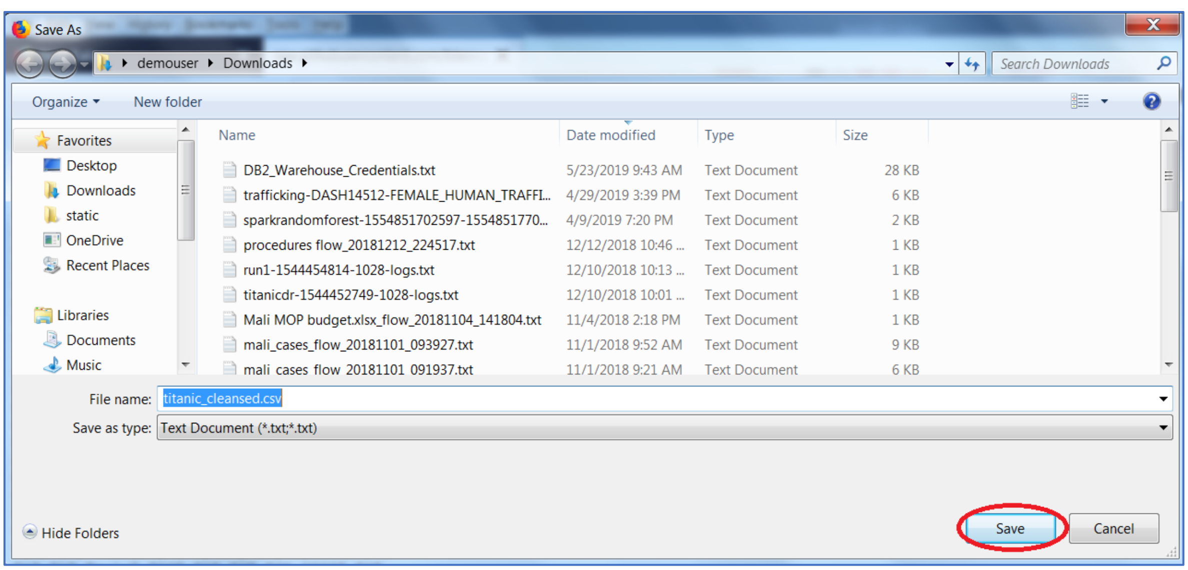
Step 1: Download the titanic_cleansed.csv data set

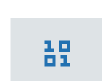
1. Download the **titanic_cleansed.csv** data file from the following location by clicking on the link here. Note this is a different file than used in the previous labs.

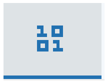
2. Right-click on the window, and click **Save Page As…**

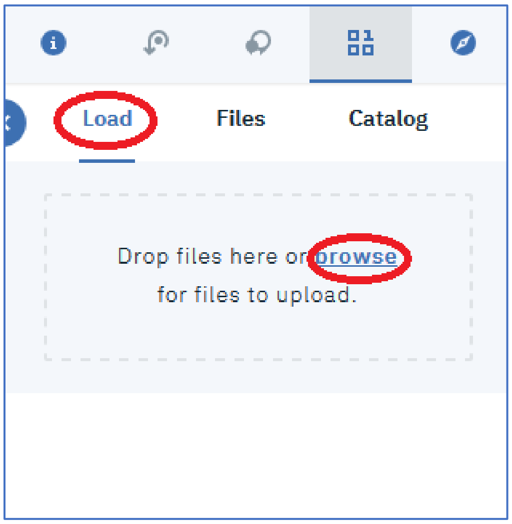


3. Click on **Save**. Note, if the file is named titanic_cleansed.csv.txt, change it to be titanic_cleansed.csv.

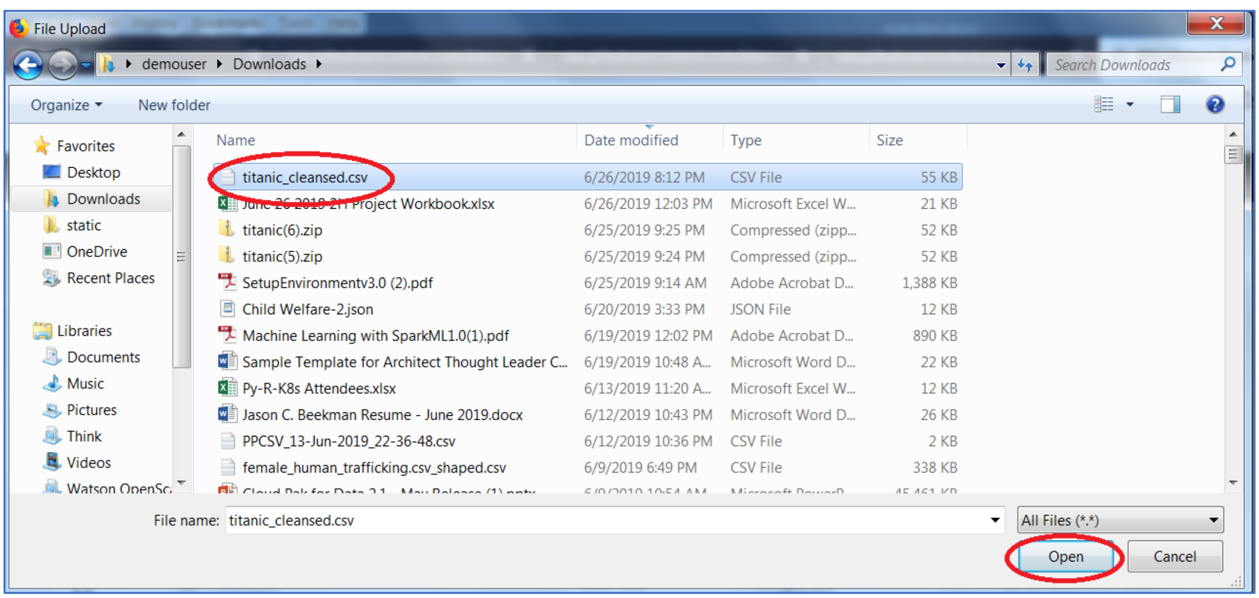


4. Go back to your Watson Studio Labs project. Click on the  icon.

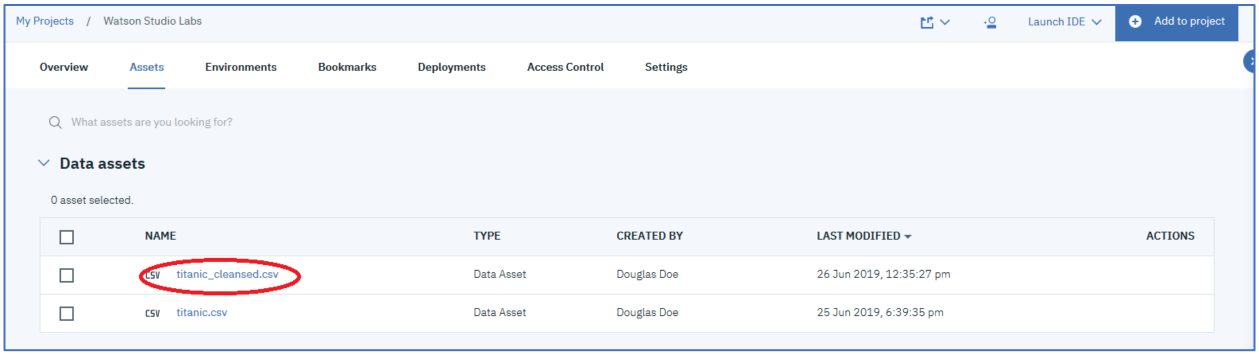
5. Click on the **Load** tab and then click on **browse**. If you don't see the **Load** tab, click on the  icon again.



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

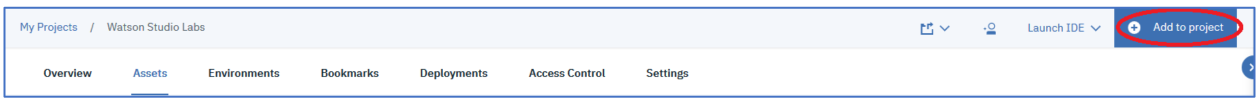


7. The titanic_cleansed.csv file is now added as a Data Asset.

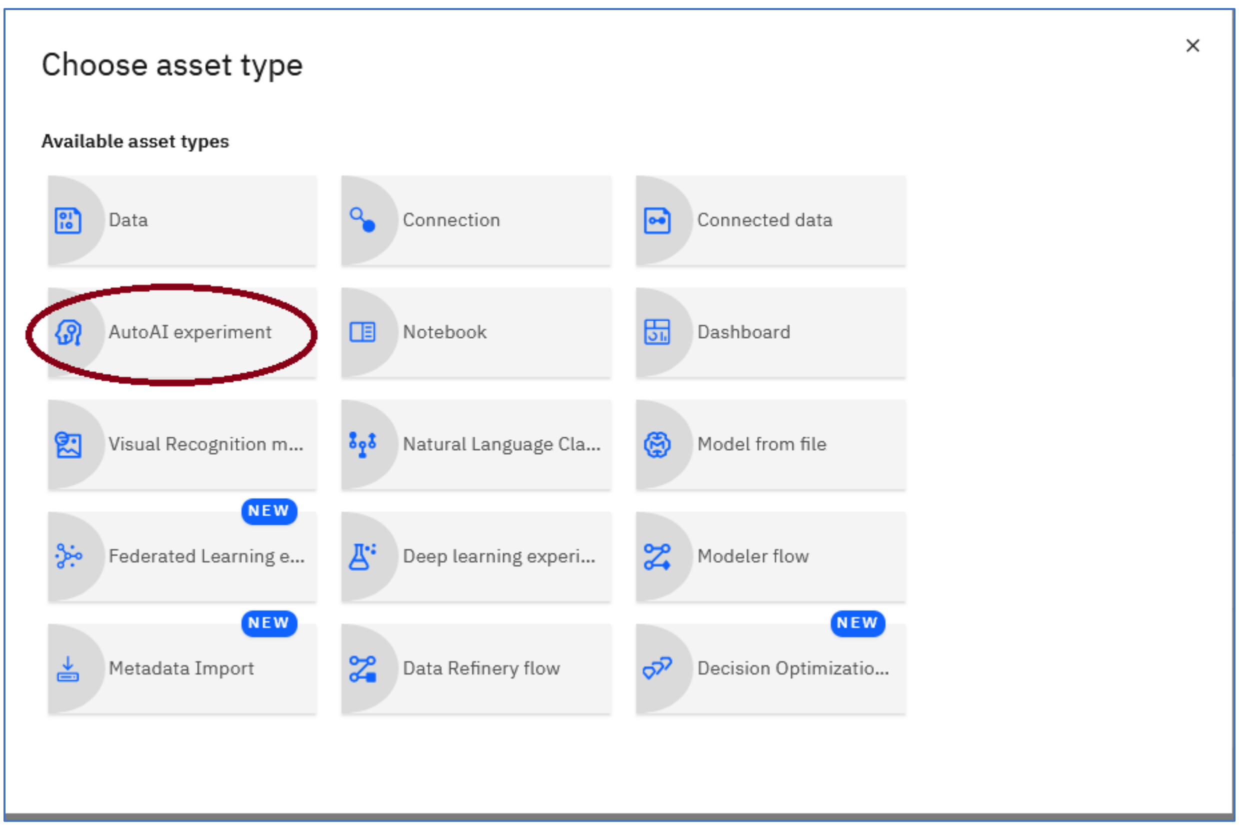


Step 2: Add an AutoAI Experiment

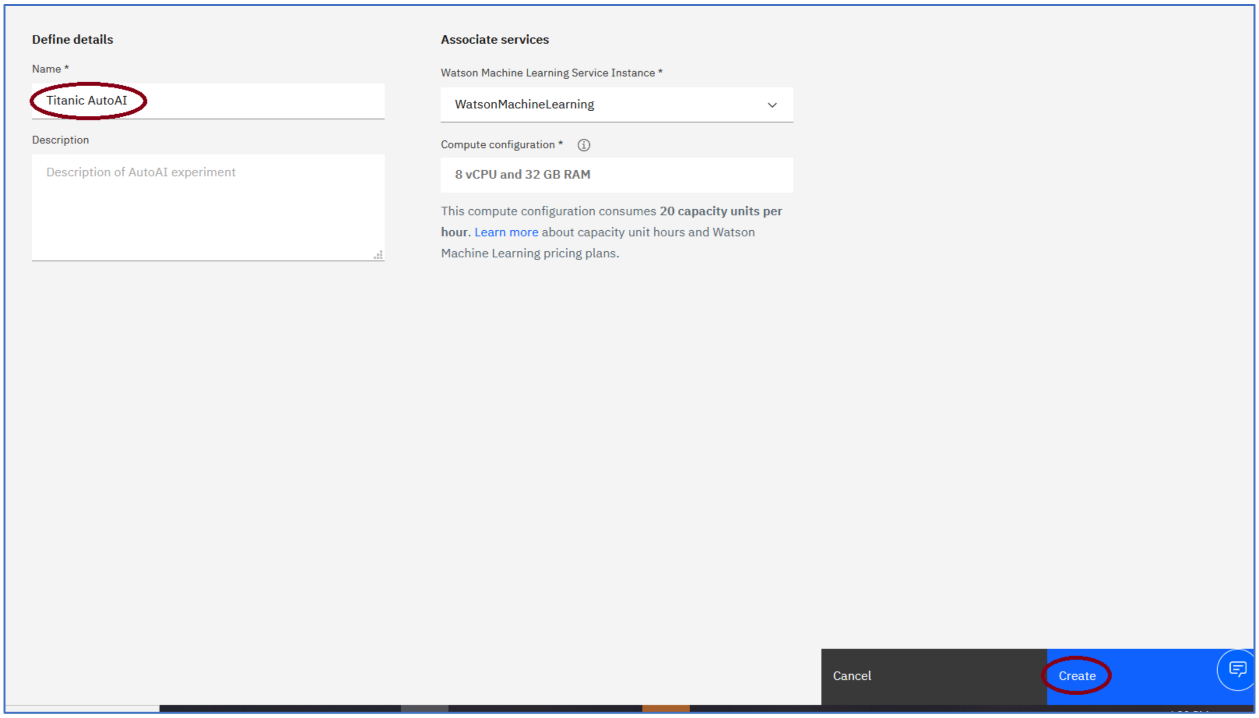
1. Click on **Add to project**.



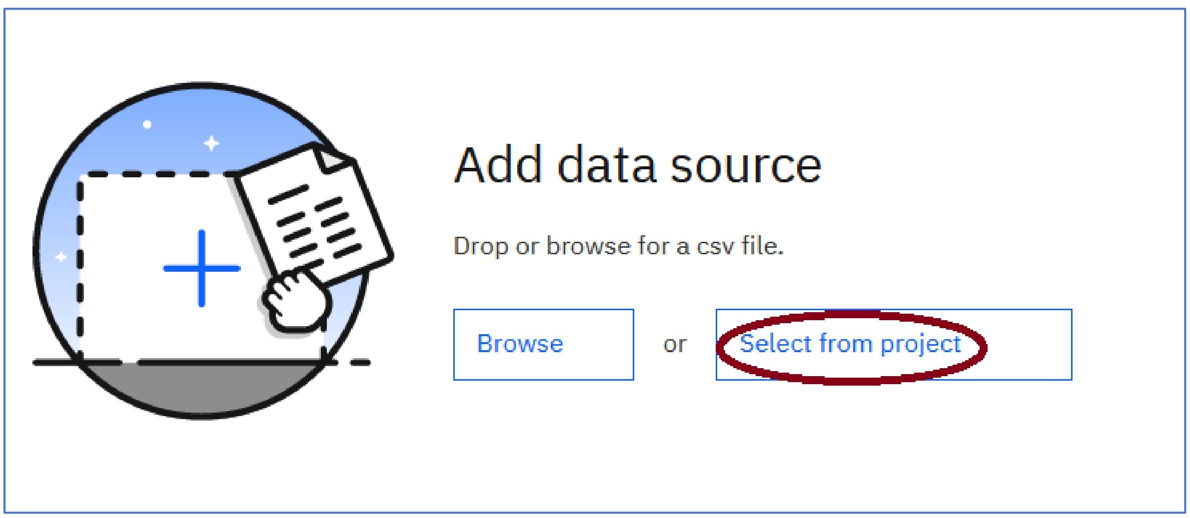
2. Click on **AutoAI experiment**



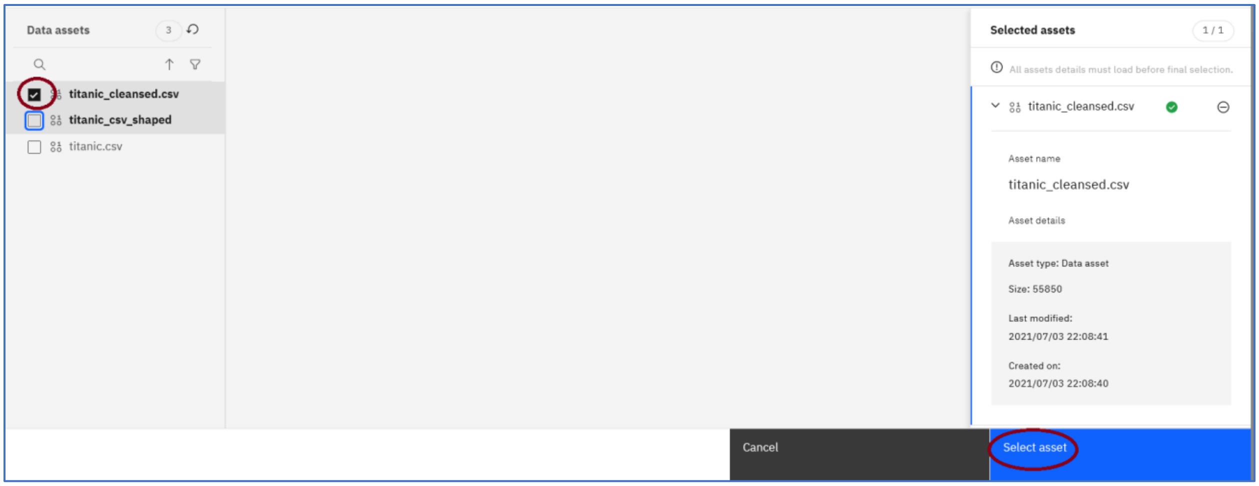
3. Enter an **Asset name**, leave the defaults for the **Watson Machine Learning** and **Compute configuration** and click on **Create**.



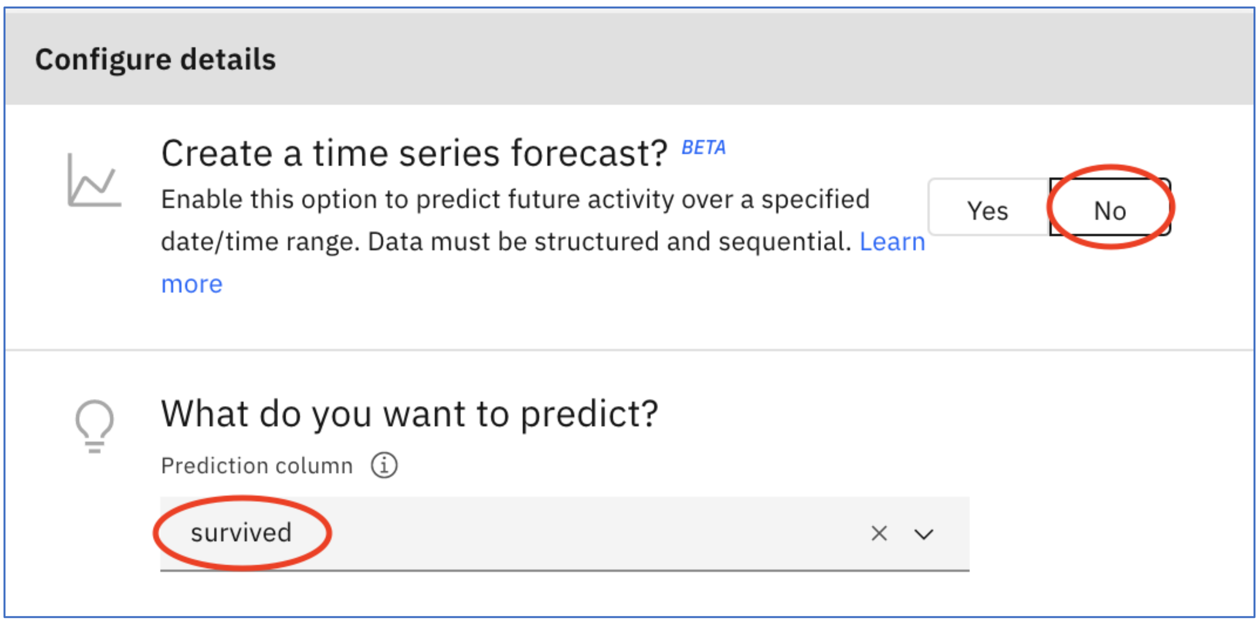
4. Click on **Select from project**.



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

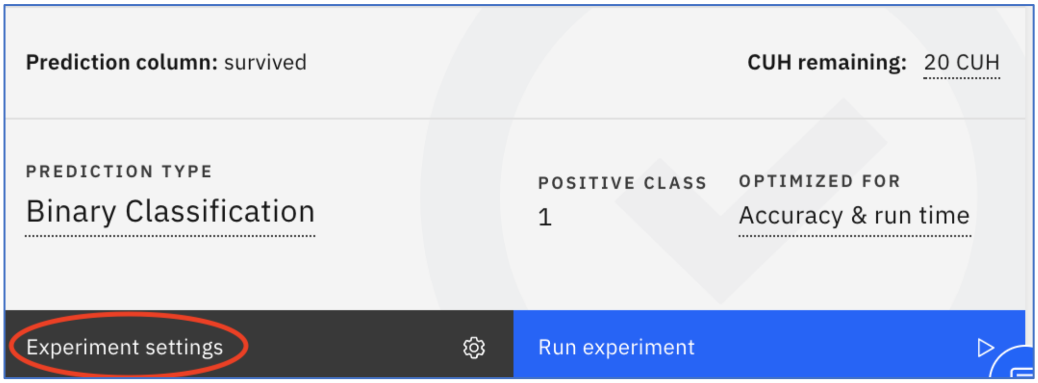


6. Select **No** when prompted on whether to **create a time series forecast** and click on **survived** as **the column to predict**.

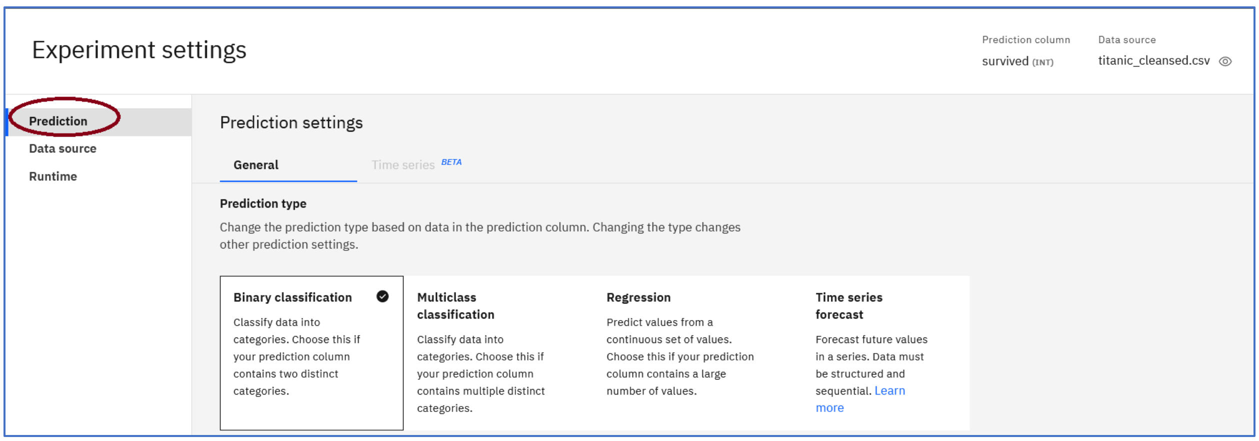


7. Note, based on this selection, the **Prediction Type** is **Binary Classification**, and the **Optimized Metric** is **Accuracy**. Further note, the **Positive Class** is correctly defaulted as “1” – survived.

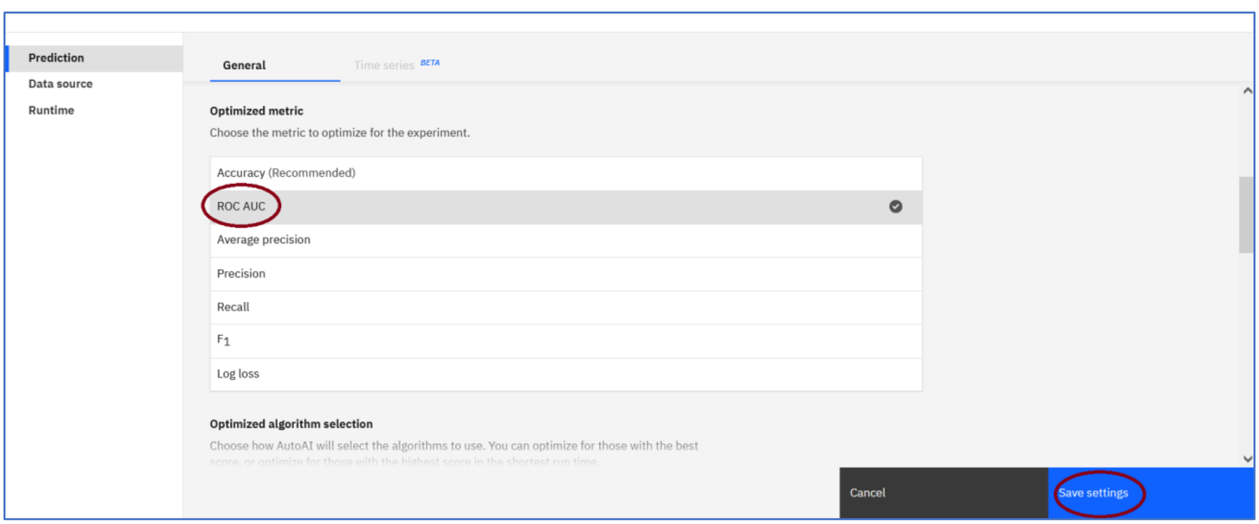
8. Click on **Experiment settings** to change the default optimized metric.



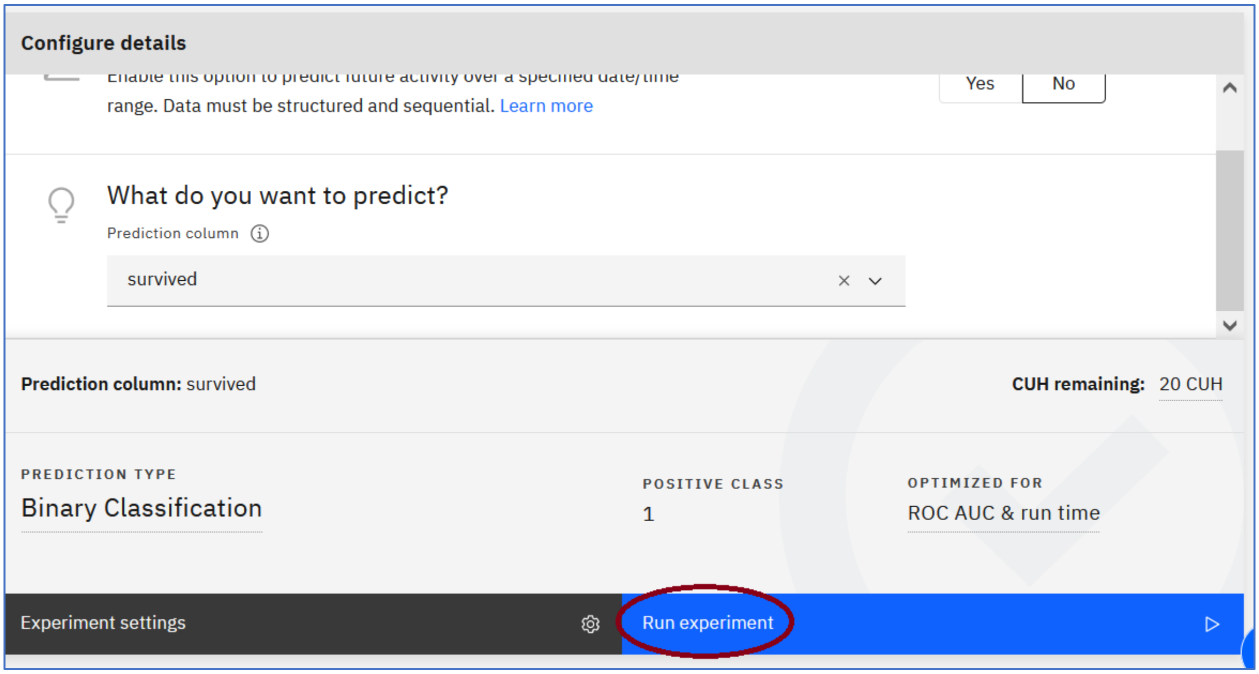
9. Click on **Prediction**.



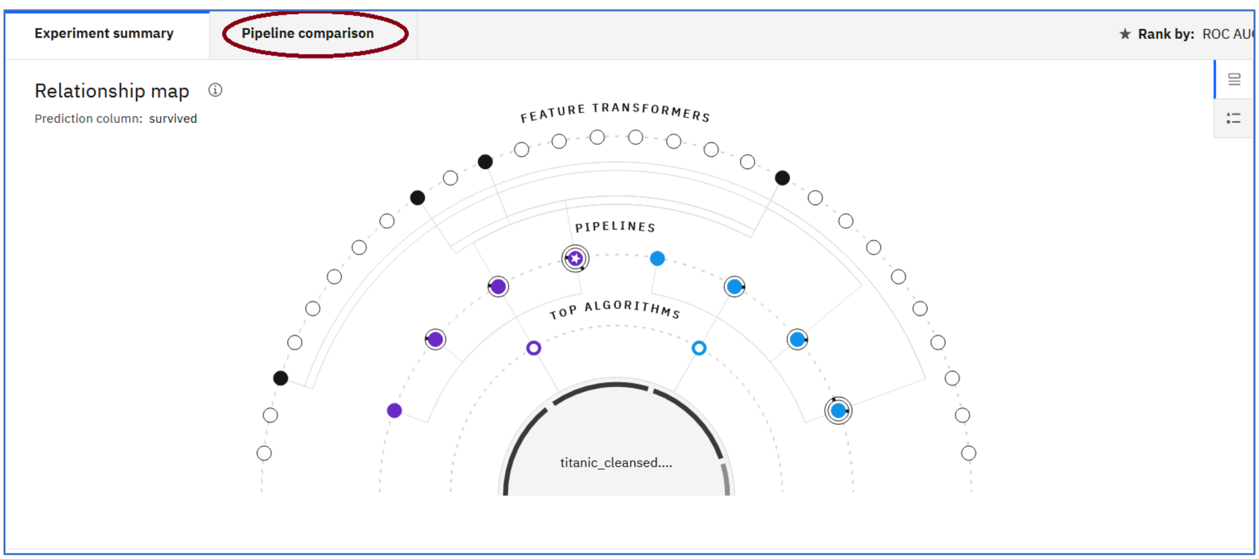
10. Ensure that the **Prediction type** is set to **Binary classification**, the **Positive class** is set to **1**. Then, scroll down and click on **ROC AUC** (Receiver Operating Characteristic Area Under the Curve) and then click on **Save Settings**



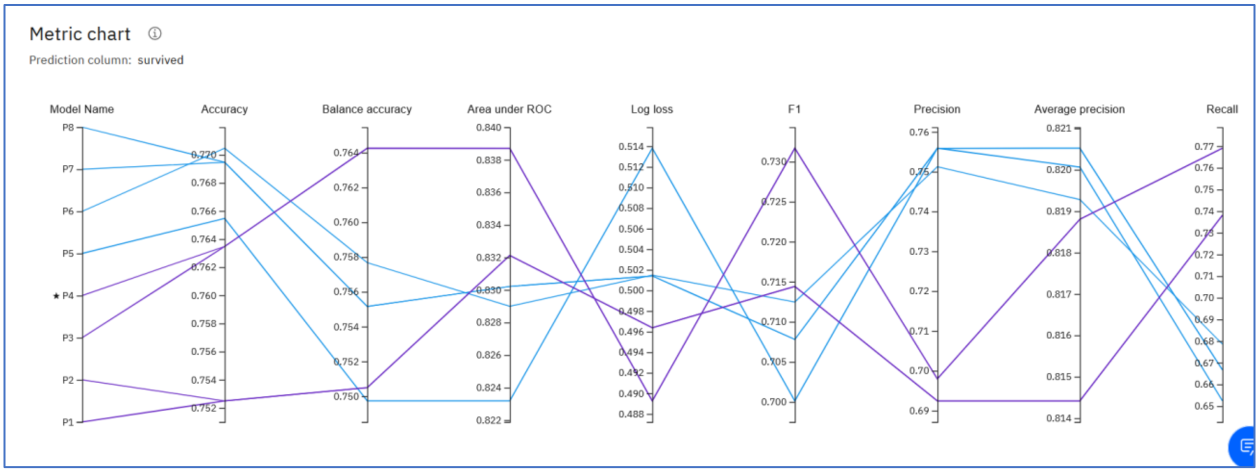
11. Click on **Run experiment**.



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. Click on **Pipeline comparison.**



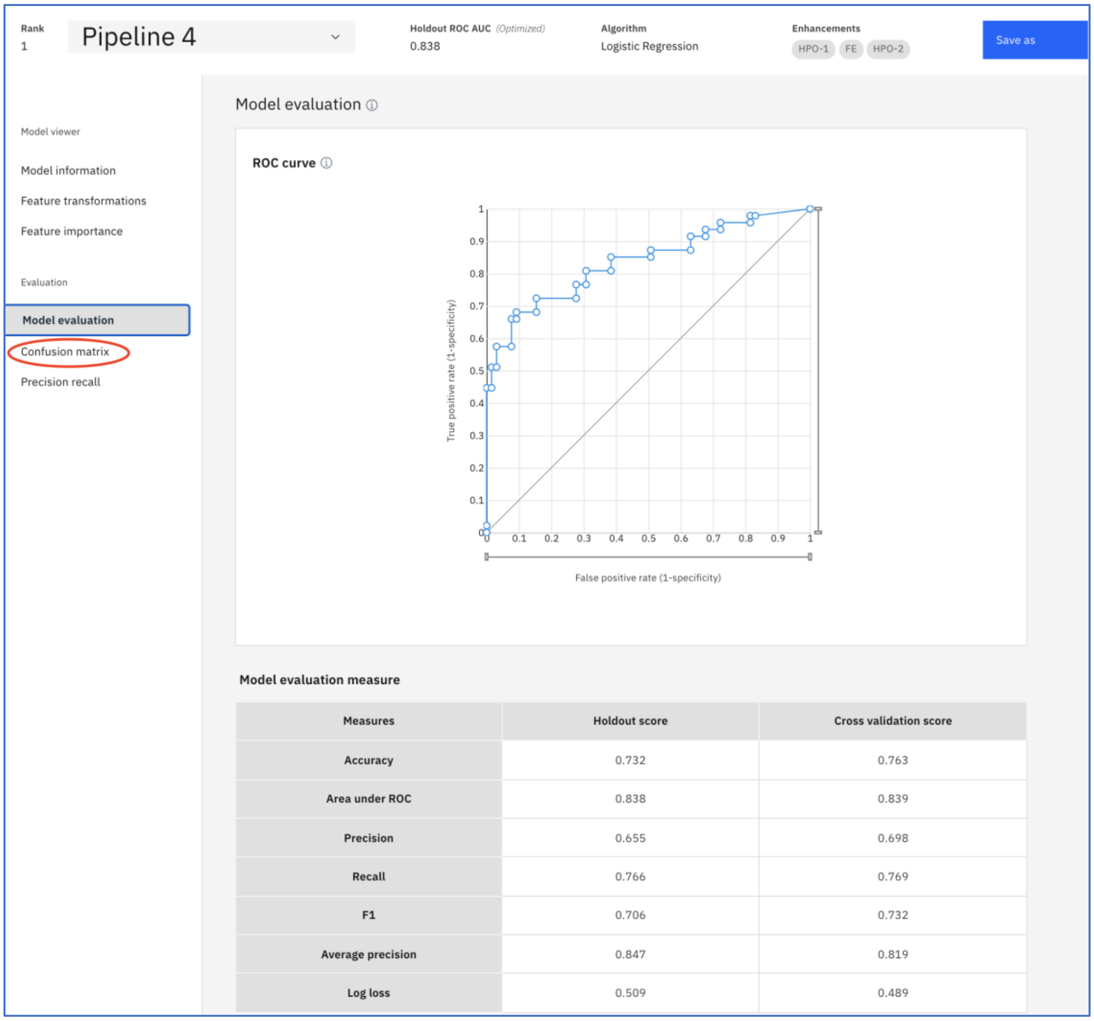
13. A comparison of all the pipelines against several performance metrics is displayed.



14. 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) Cross Validation	Enhancements	Build time
★	1	Pipeline 4	Logistic Regre...	0.839	HPO-1 FE HPO-2	00:00:10
	2	Pipeline 3	Logistic Regre...	0.839	HPO-1 FE	00:00:20
	3	Pipeline 1	Logistic Regre...	0.832	*None*	00:00:01
	4	Pipeline 2	Logistic Regre...	0.832	HPO-1	00:00:03
	5	Pipeline 7	XGB Classifier	0.830	HPO-1 FE	00:00:27
	6	Pipeline 8	XGB Classifier	0.830	HPO-1 FE HPO-2	00:00:22
	7	Pipeline 6	XGB Classifier	0.829	HPO-1	00:00:07
	8	Pipeline 5	XGB Classifier	0.823	*None*	00:00:01

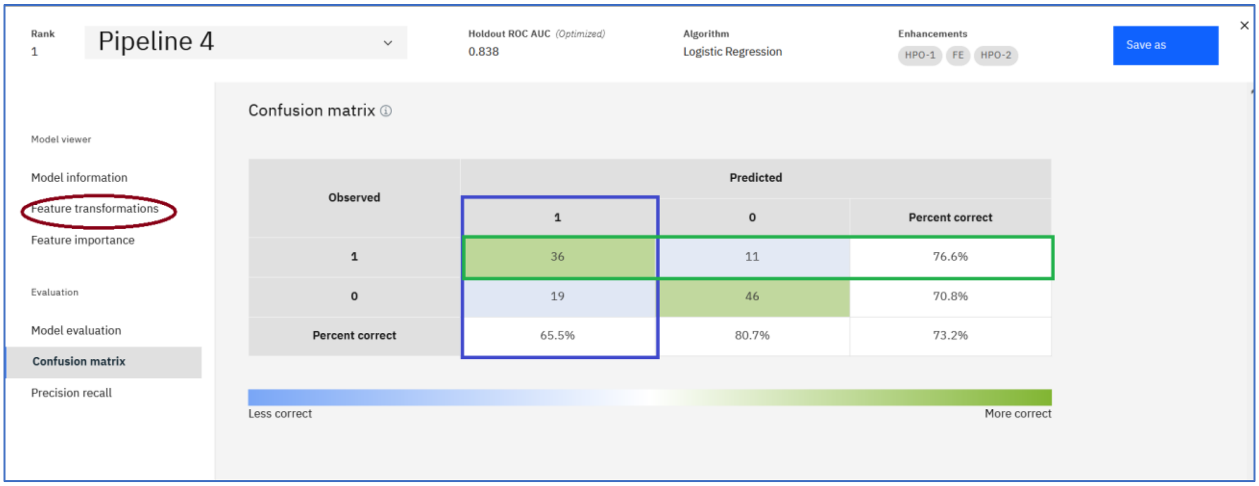
15. Metrics are displayed for both the holdout sample and the training sample (cross-validation). Click on **Confusion Matrix**.



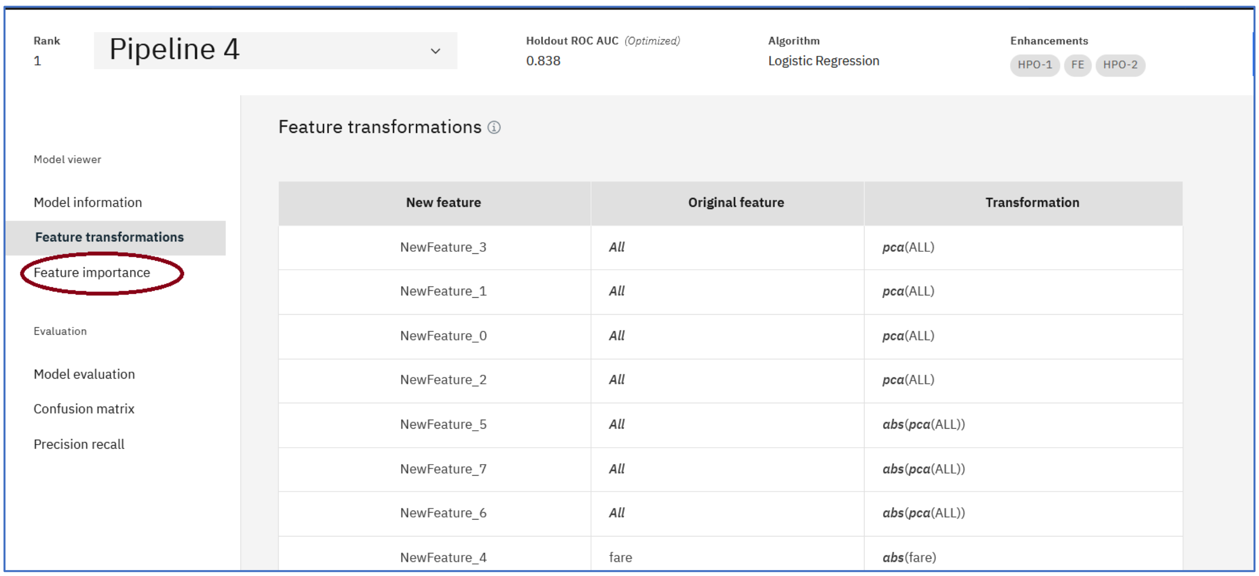
16. The Confusion Matrix is displayed for the holdout sample. The different metrics are computed based on the numbers in the Confusion Matrix. For example, Precision is defined by the percentage of predicted positives that are actually positive (i.e. the percentage of predicted survivors that survived).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

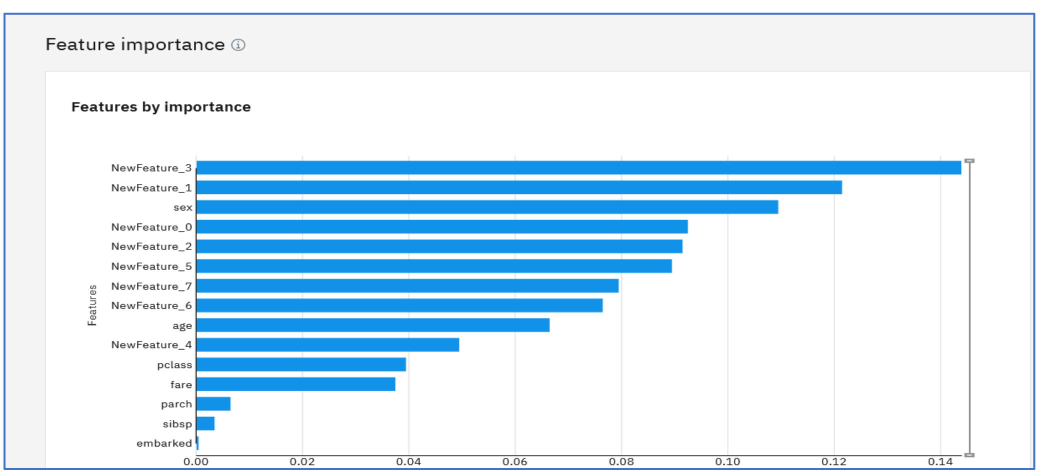
    Recall = True Positive/(True Positive + False Negative) - shown inside green rectangle on diagram below. This is also called the True Positive Rate. After viewing the Confusion Matrix, click on the **Feature Transformation** option.



17. Eight new features are derived as shown below. Click on **Feature Importance**.

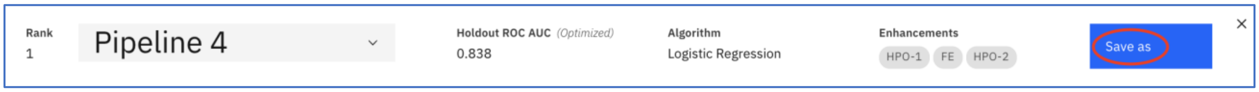


18. According to the Feature Importance, two of the derived features are the most important followed by the sex input feature.

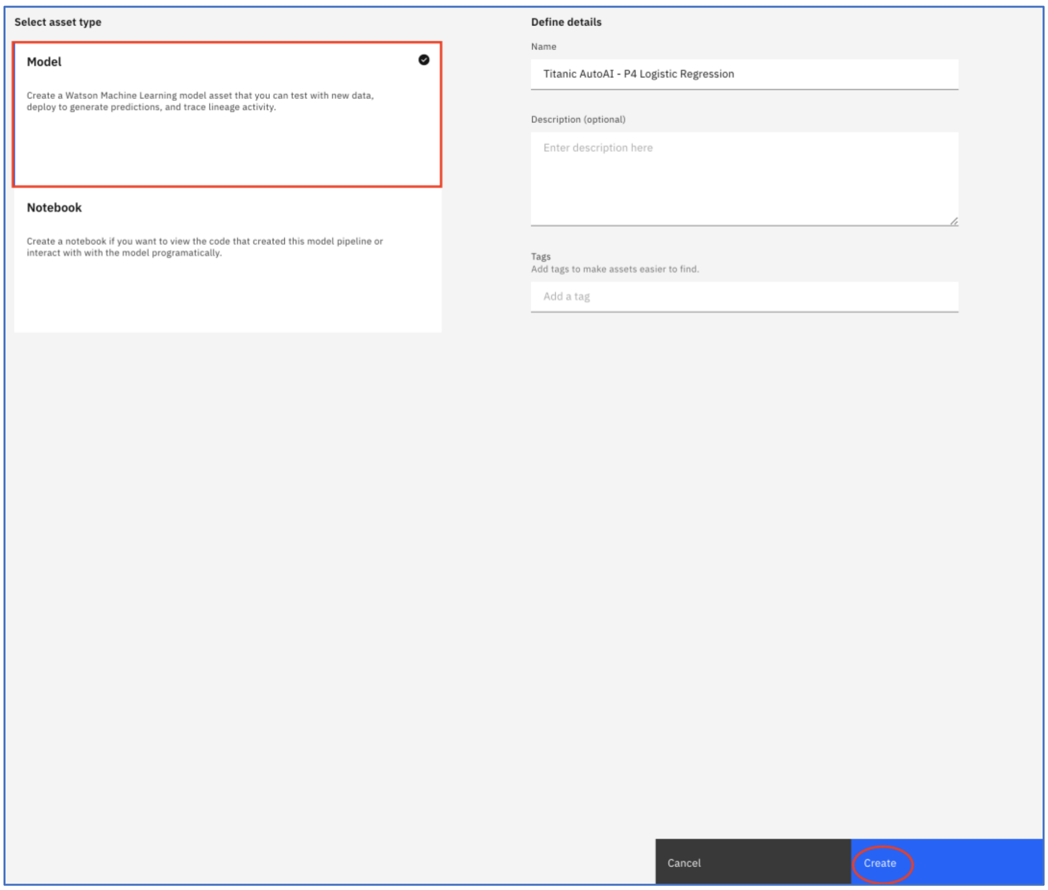


Step 3 – Save and Deploy the Selected Model

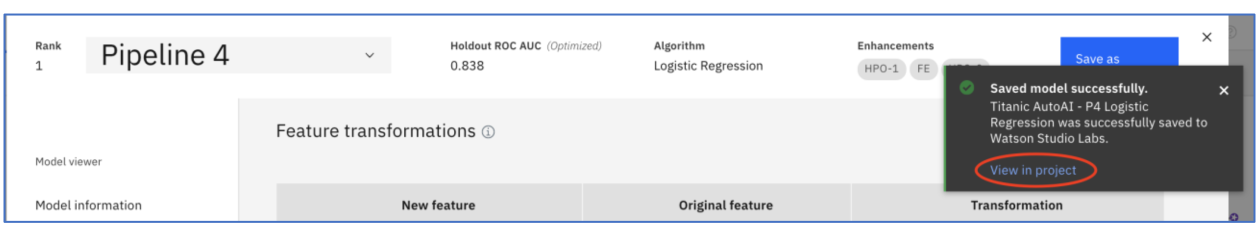
1. Click on **Save as**.



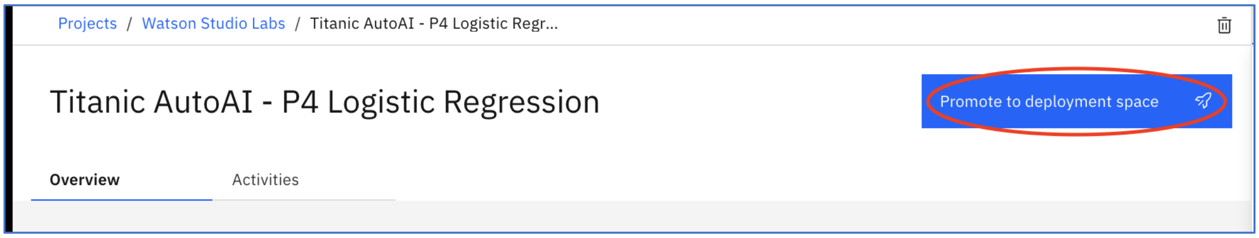
2. You have the option to save a Model or a Notebook. The notebook contains the code used to generate the pipeline. In this way a data scientist could use this as a starting point to tune the model even further. We will save the model. Optionally change the default name and click on **Save**.



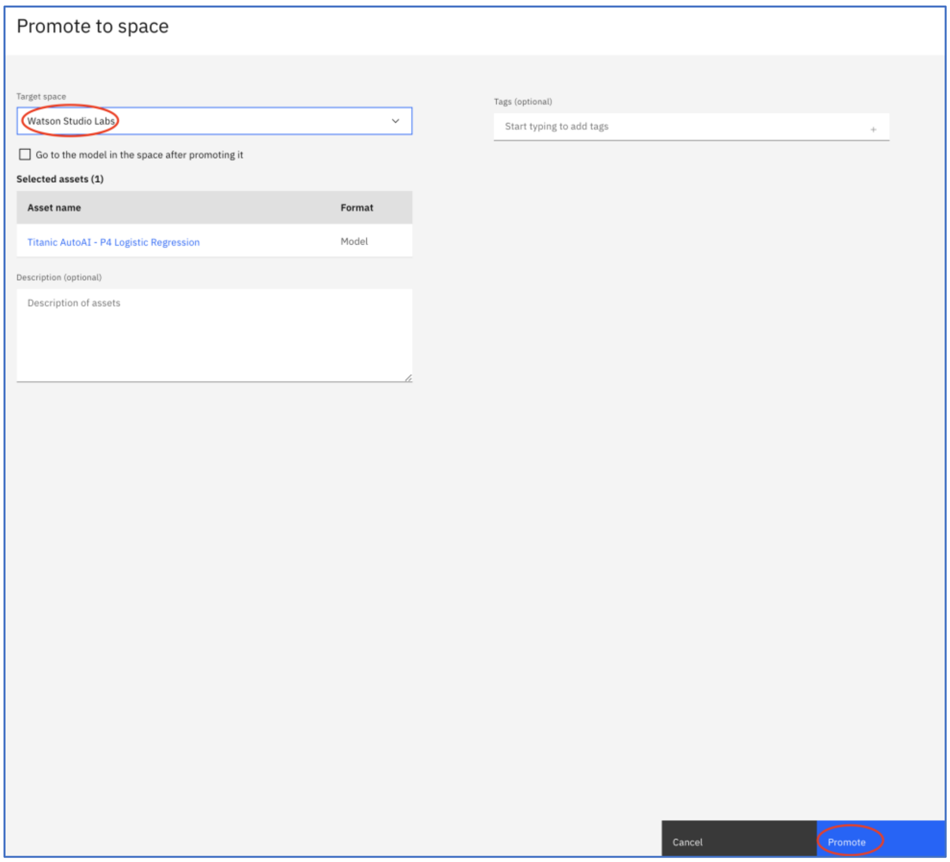
3. The model is successfully saved. Click on **View in Project**.



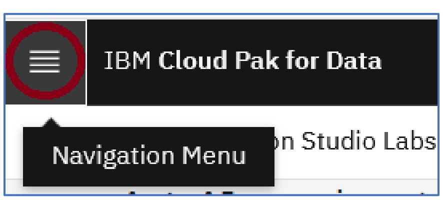
4. In order to deploy the model, we need to promote it to the deployment space that was created in Lab-1. Click on **Promote to deployment space**.



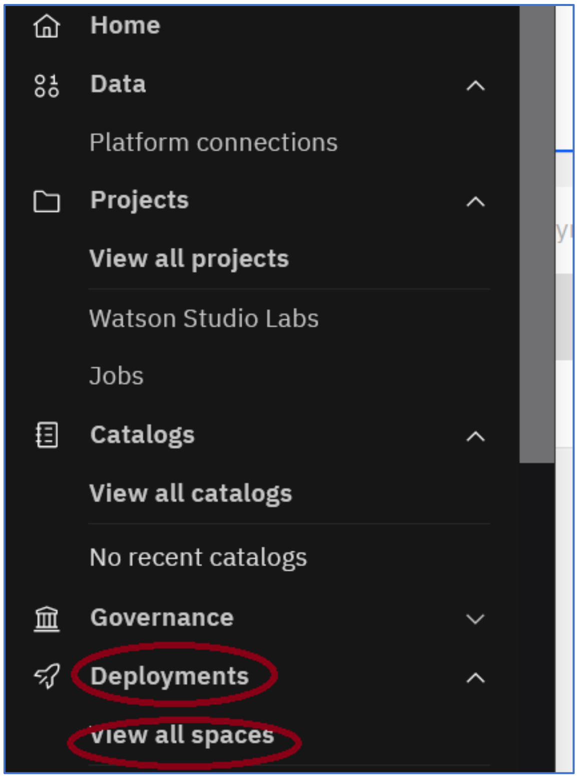
5. Set the **Target space** to **Watson Studio Labs** and click on **Promote**.



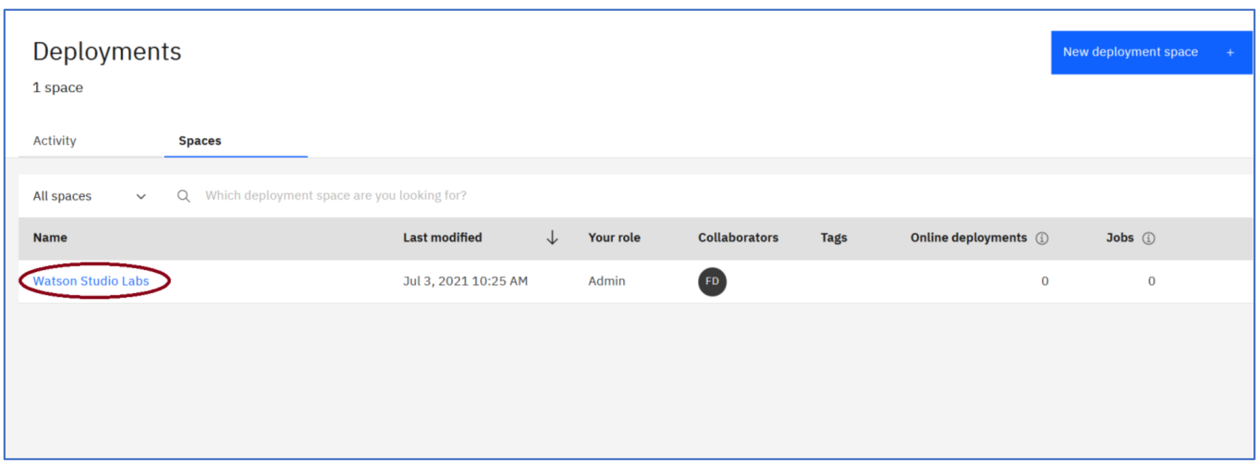
6. Click on the hamburger icon ☰.



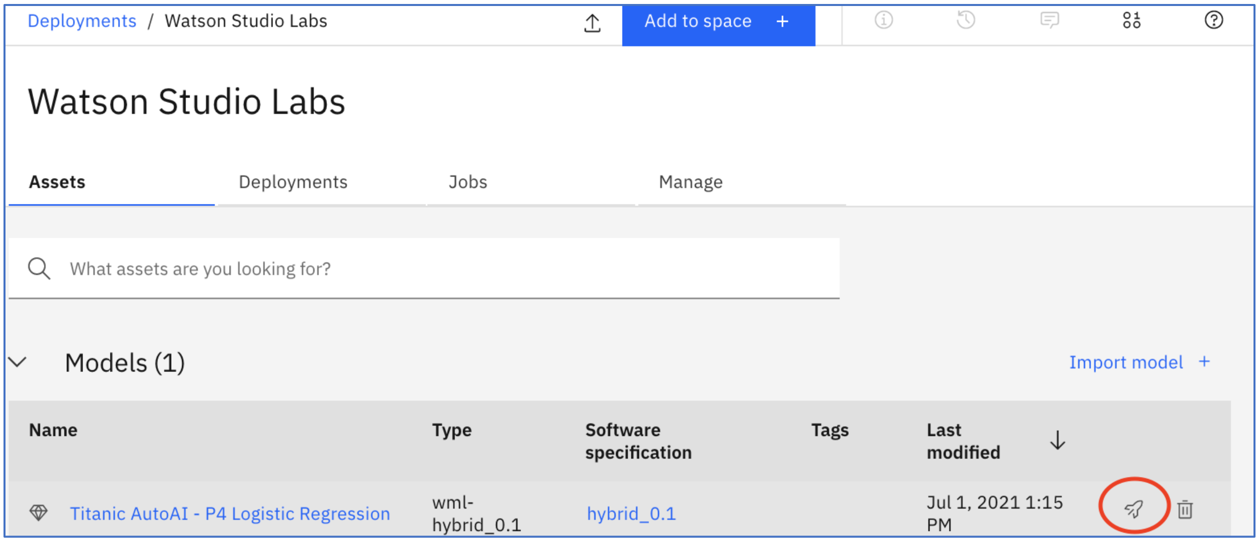
7. Click on **Deployments** and **View All Spaces.**



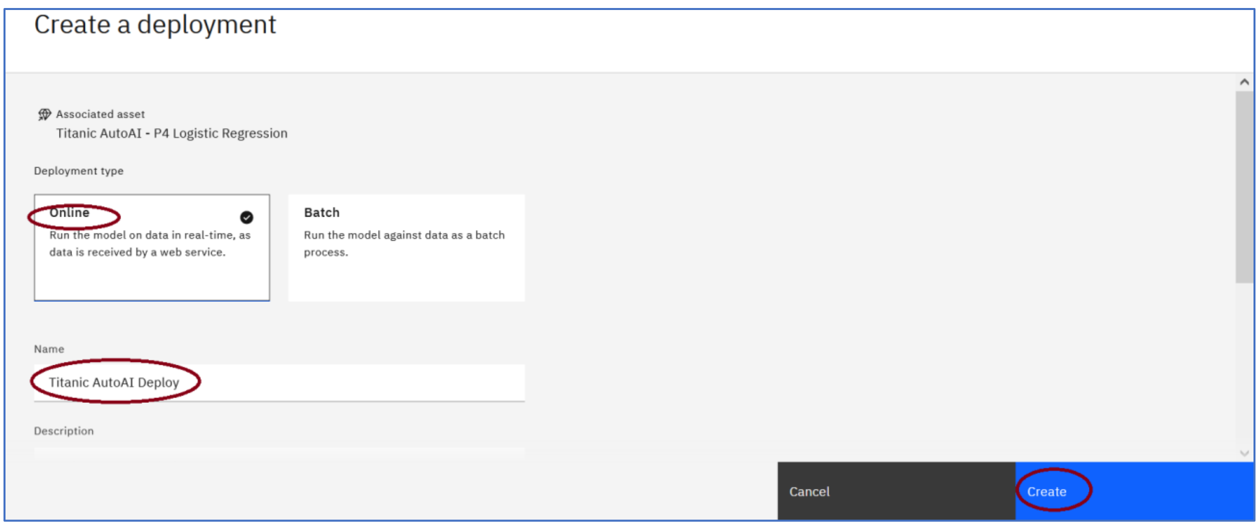
8. Click on **Watson Studio Labs**.



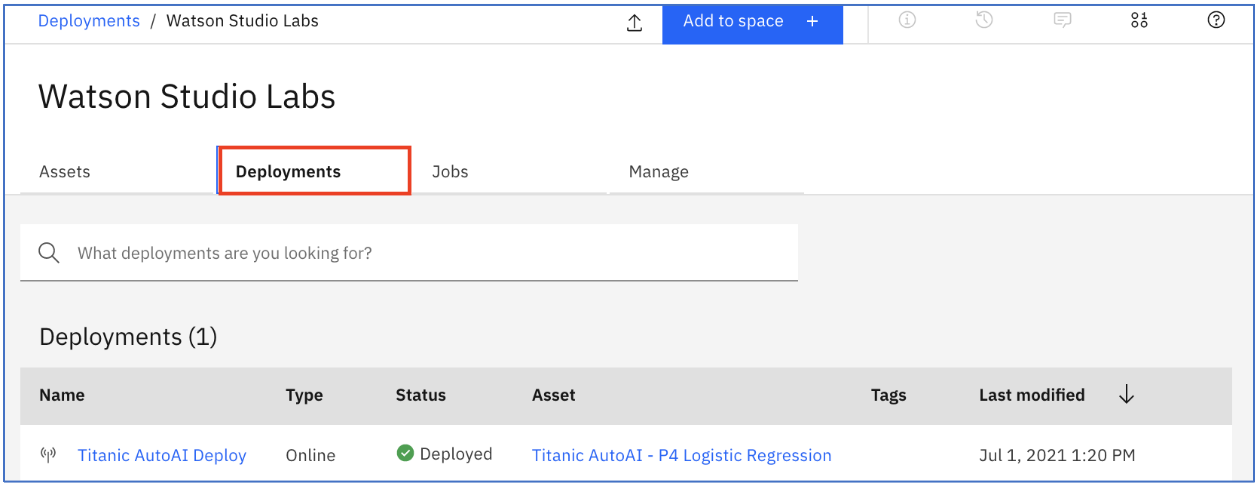
9. Hover over the model and on the right the Deploy icon  will appear. Click on the icon.



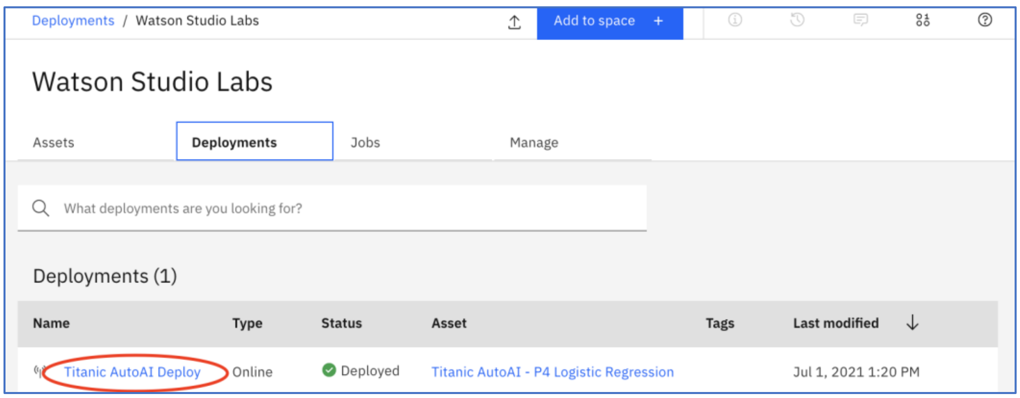
10. Click on **Online**, enter **Titanic AutoAI Deploy** for the **Name** of the deployment, optionally enter a **Description**, and click **Create**.



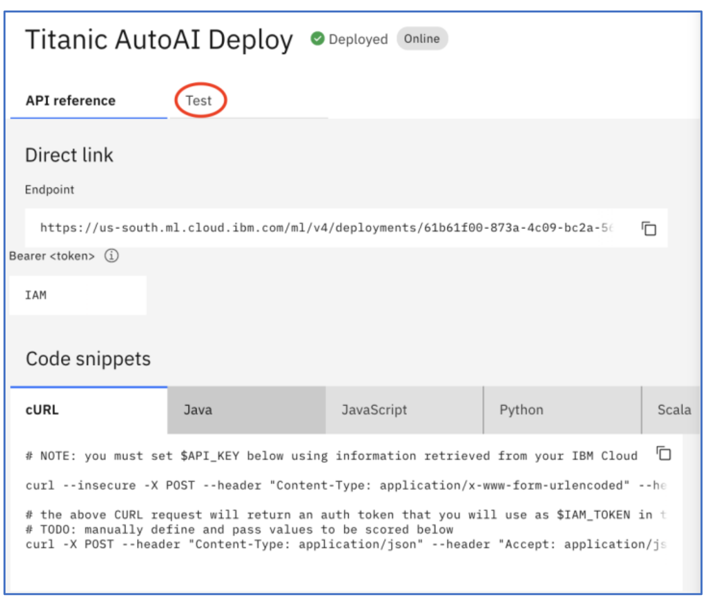
11. Click on the **Deployment** Tab to see the deployment status.



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



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 on **Test**.

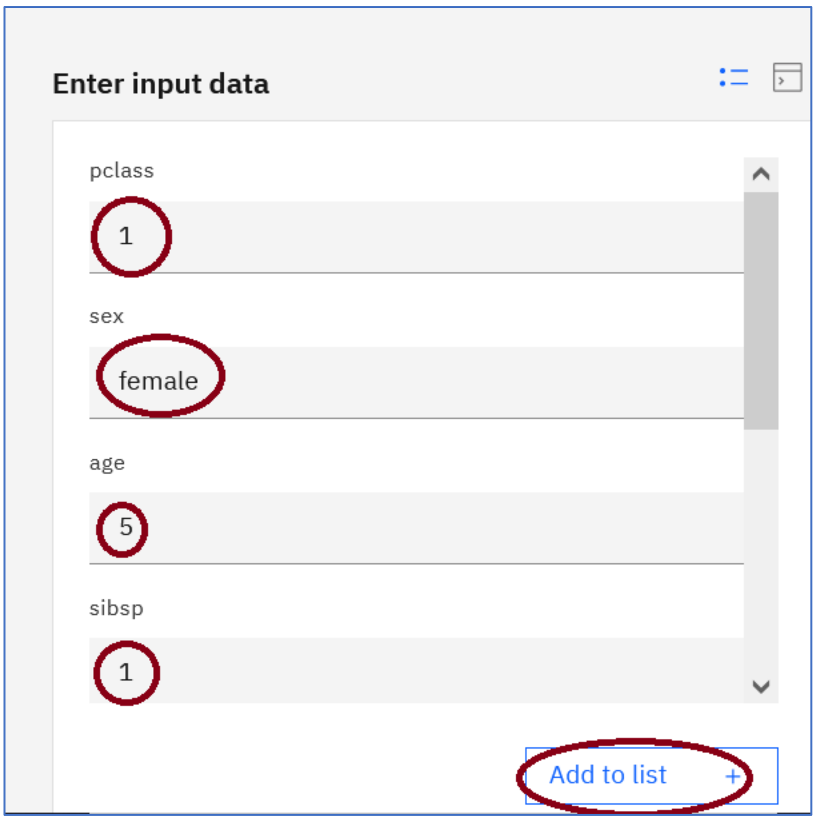


14. Enter values for a passenger. For example,
    pclass – 1
    sex – female
    age – 5
    sibsp – 1

    Scroll down to add

    parch – 2
    fare – 23
    embarked – S

    and click **Add to list.**



15. Click **Predict(1)**

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

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.