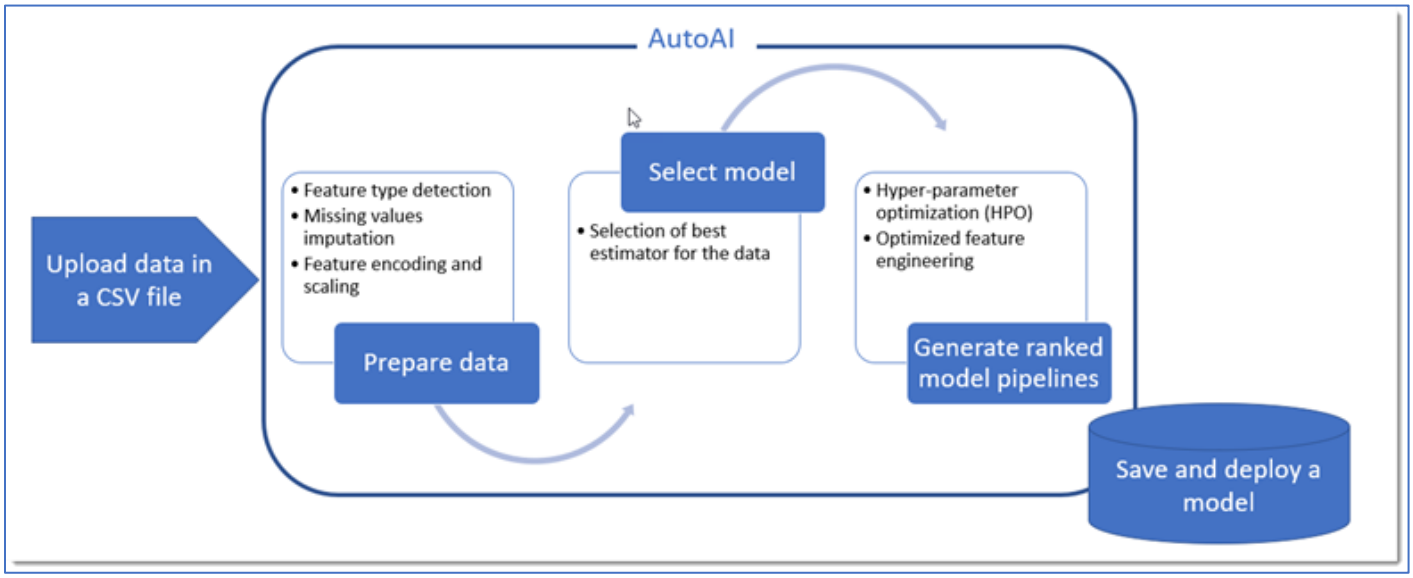
AutoAI Lab

This lab will demonstrate the new and exciting 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 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

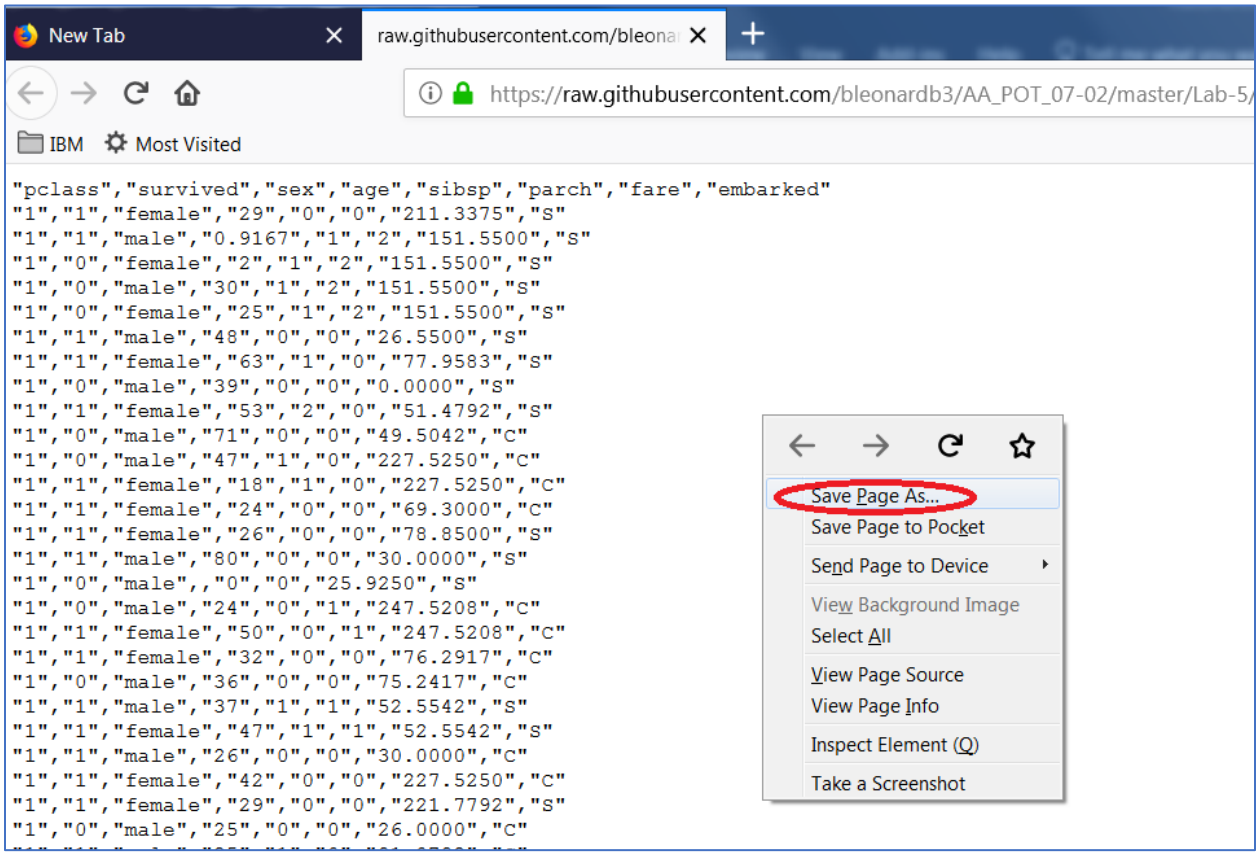
We will perform the following steps in this lab:

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

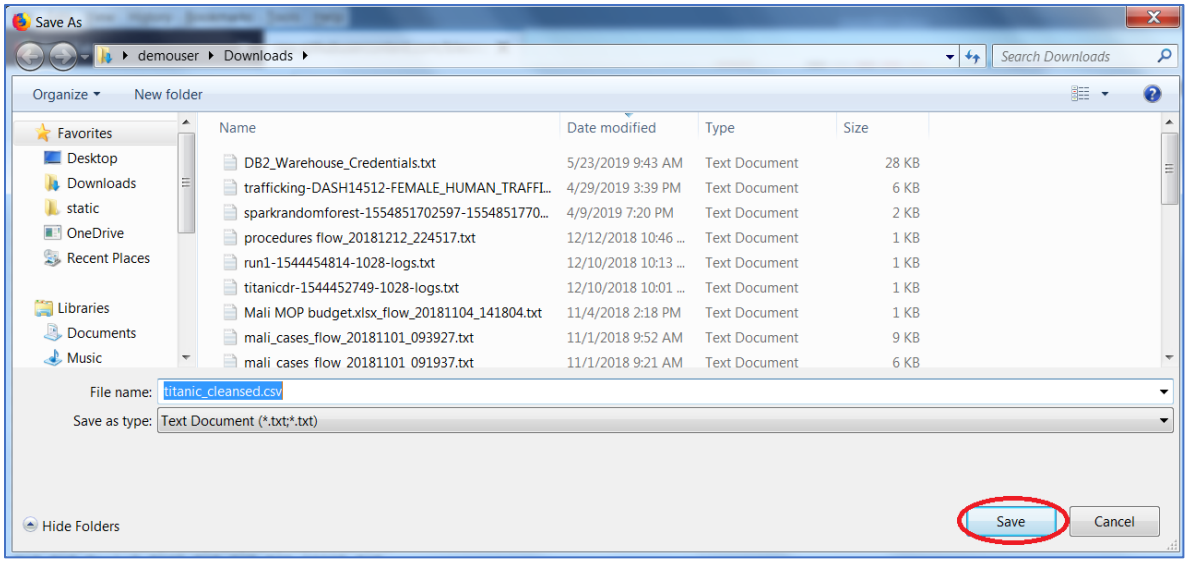
Step 1: Download the titanic_cleansed.csv data set

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…**

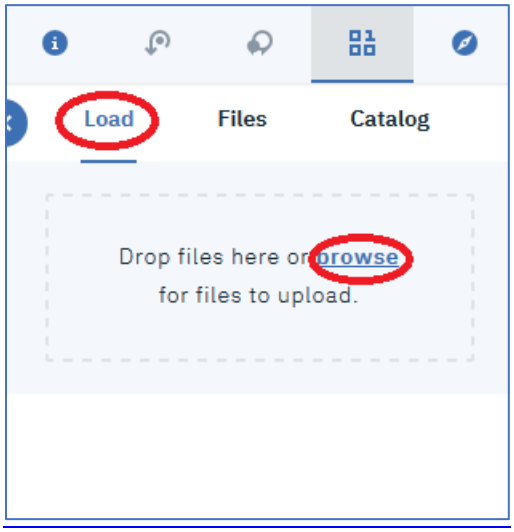


3. Click on **Save**. Note, if the file is named titanic_cleansed.csv.txt, change it to be titanic_cleansed.csv.

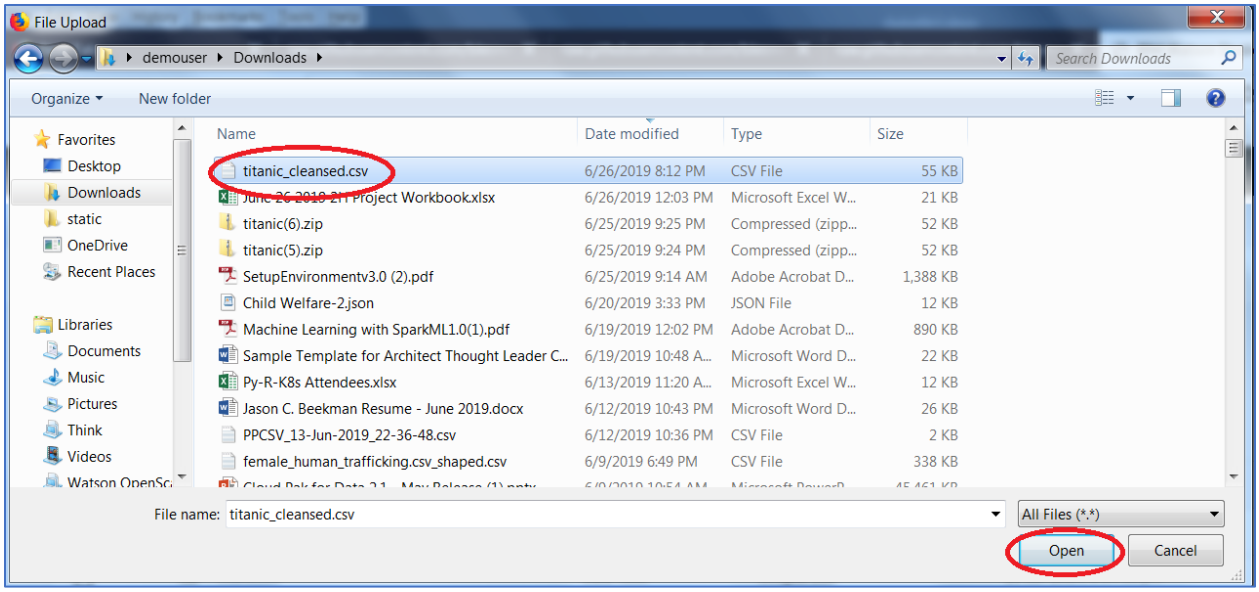


4. Go back to your Watson Studio Labs project. Click on the  icon.

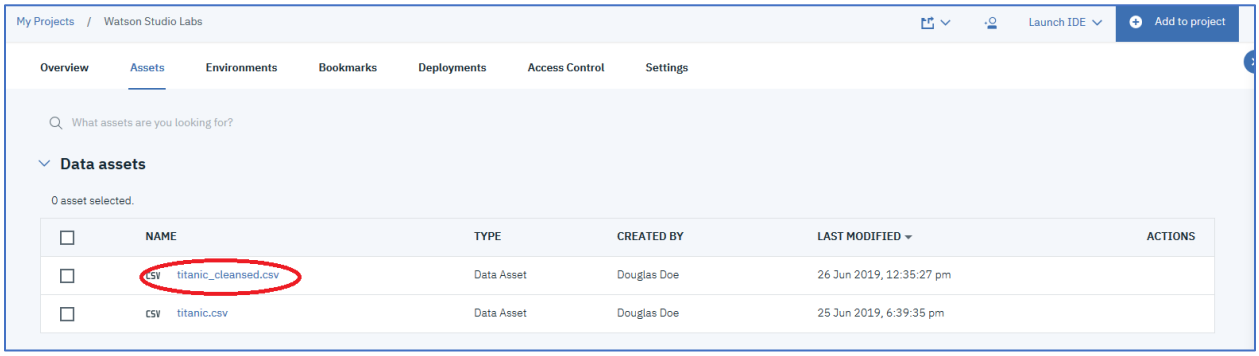
5. Click on the **Load** tab and then click on **browse**. If you don't see the **Load** tab, click on the  icon again.



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

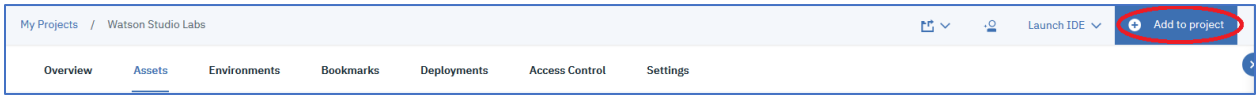


7. The titanic_cleansed.csv file is now added as a Data Asset.

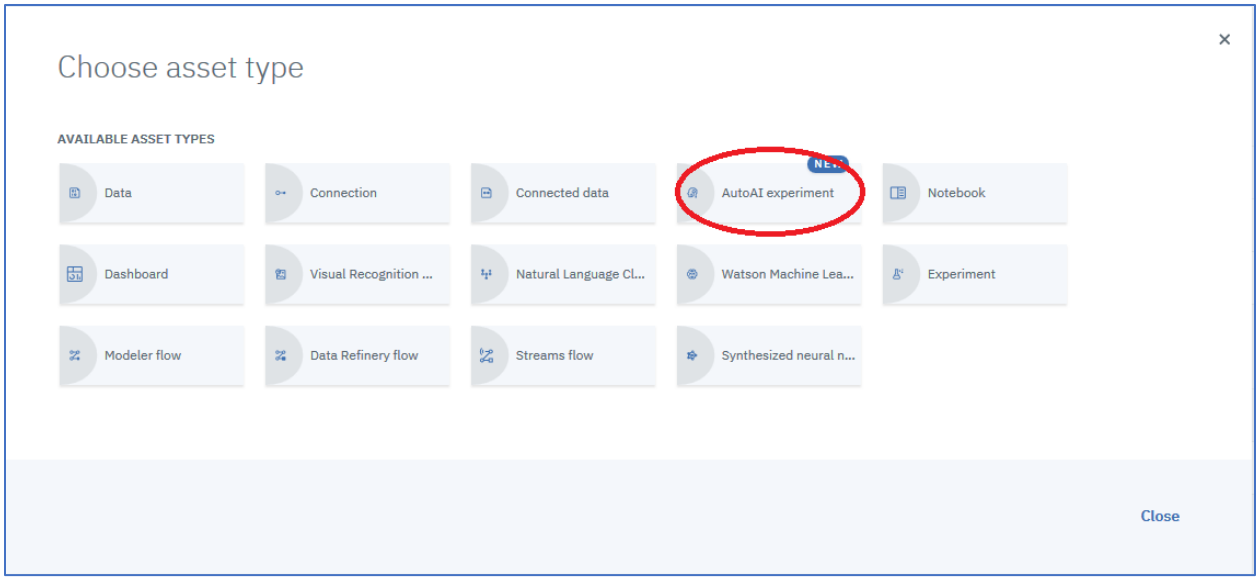


Step 2: Add an AutoAI Experiment

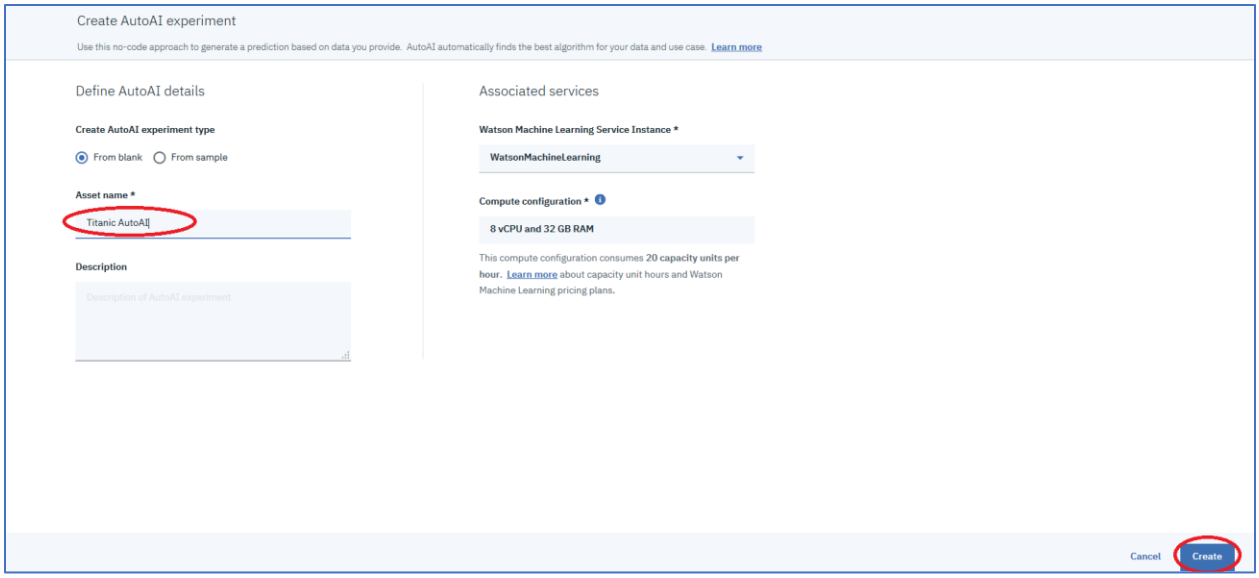
1. Click on **Add to project**.



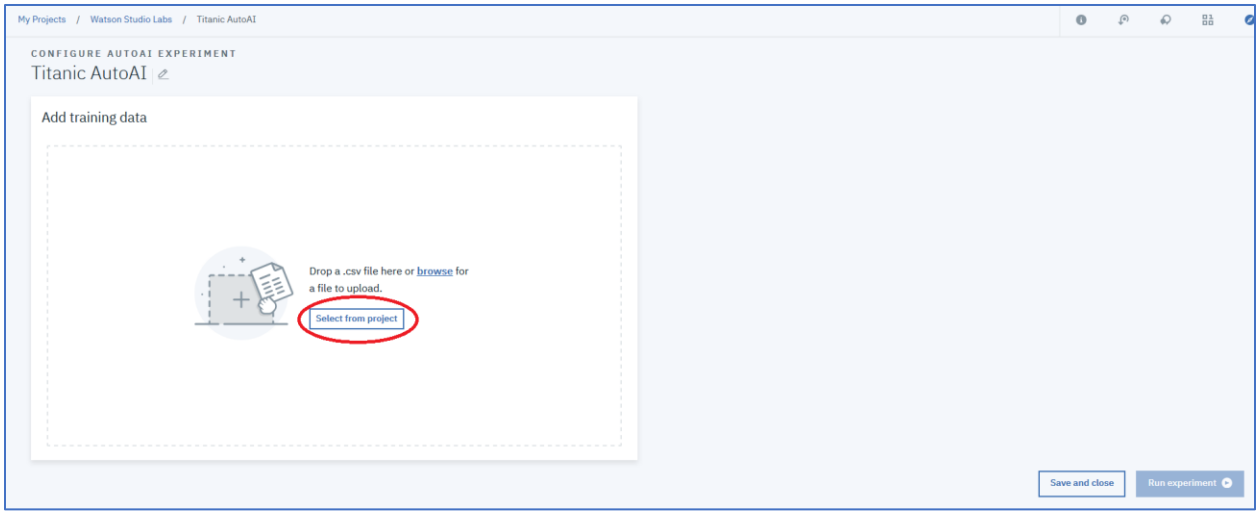
2. Click on **AutoAI experiment**



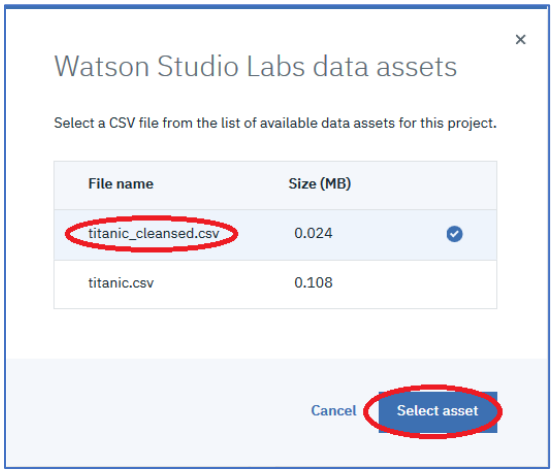
3. Enter an **Asset name**, leave the defaults for the **Watson Machine Learning** and **Compute configuration** and click on **Create**.



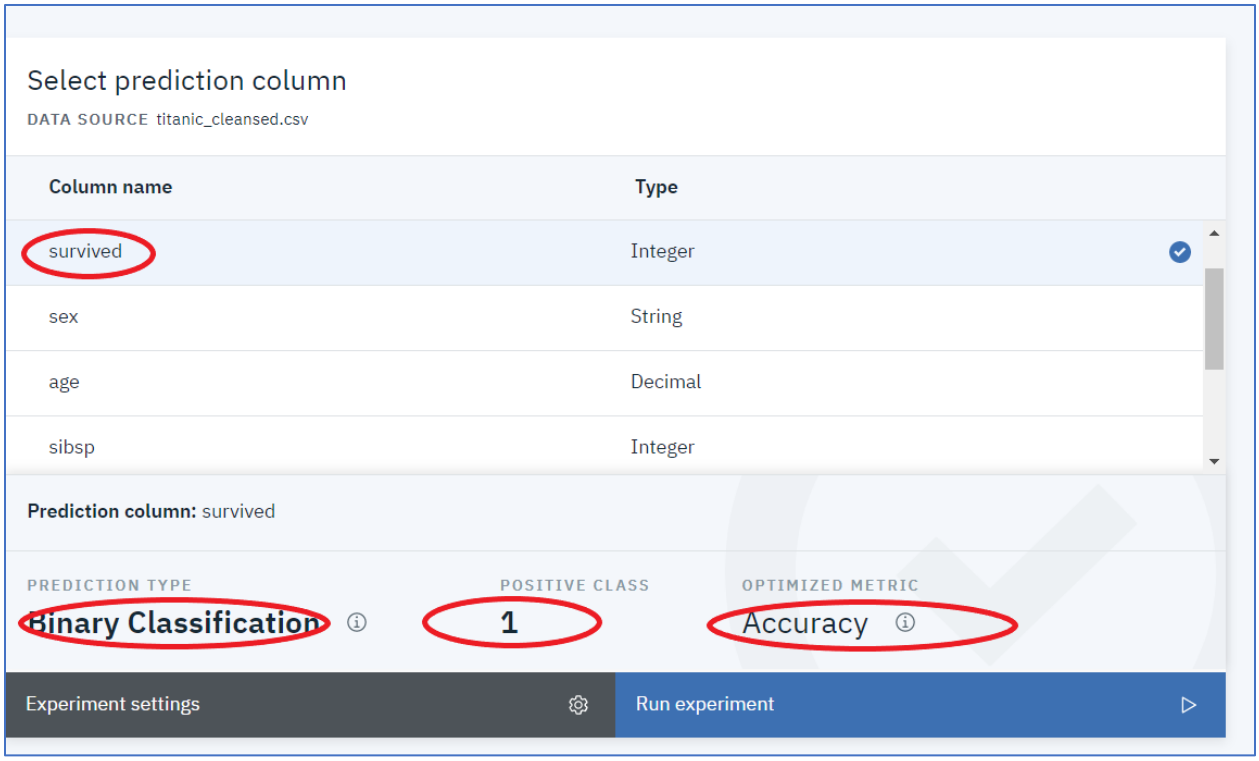
4. Click on **Select from project**.



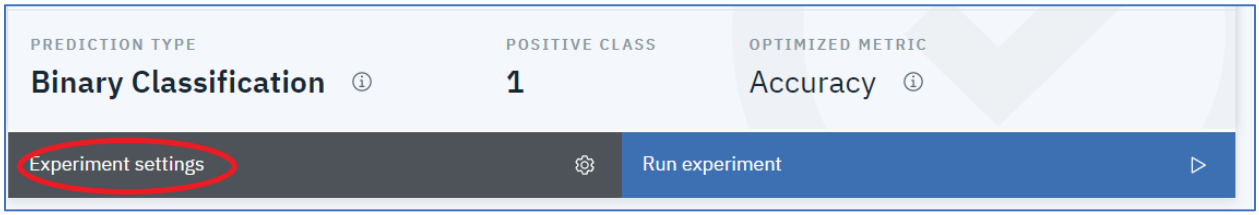
5. Click on **titanic_cleansed.csv** and then click on **Select asset**.



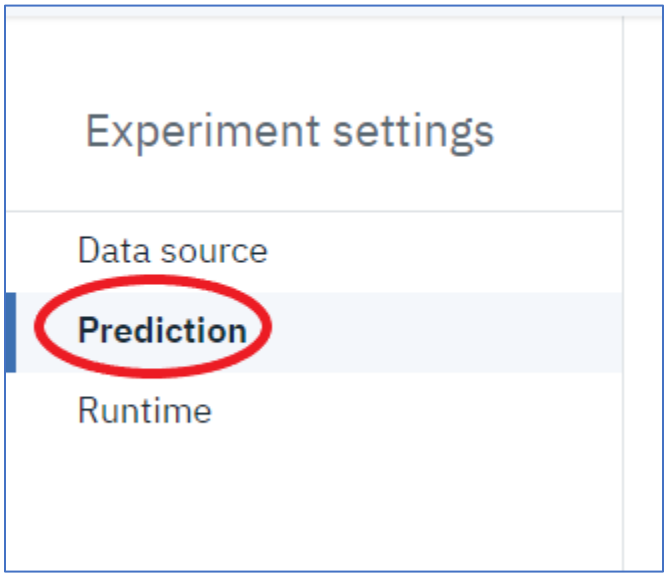
6. Click on **survived** as **the column to predict**. Note, based on this selection, the **Prediction Type** is **Binary Classification**, and the **Optimized Metric** is **Accuracy**. Further note, the **Positive Class** is correctly defaulted as "1" – survived.



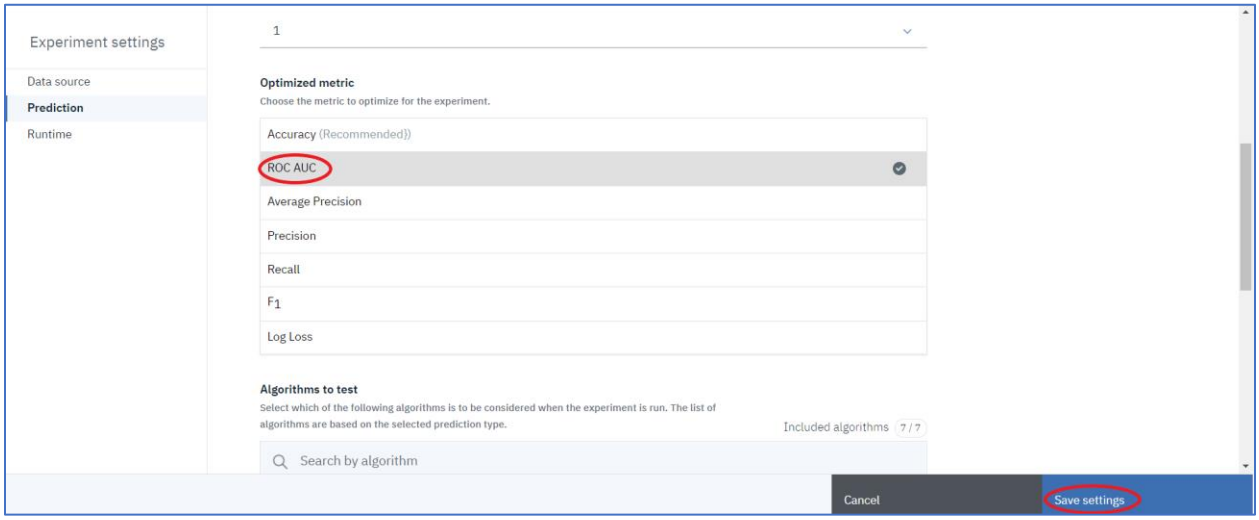
7. Click on **Experiment settings** to change the default optimized metric.



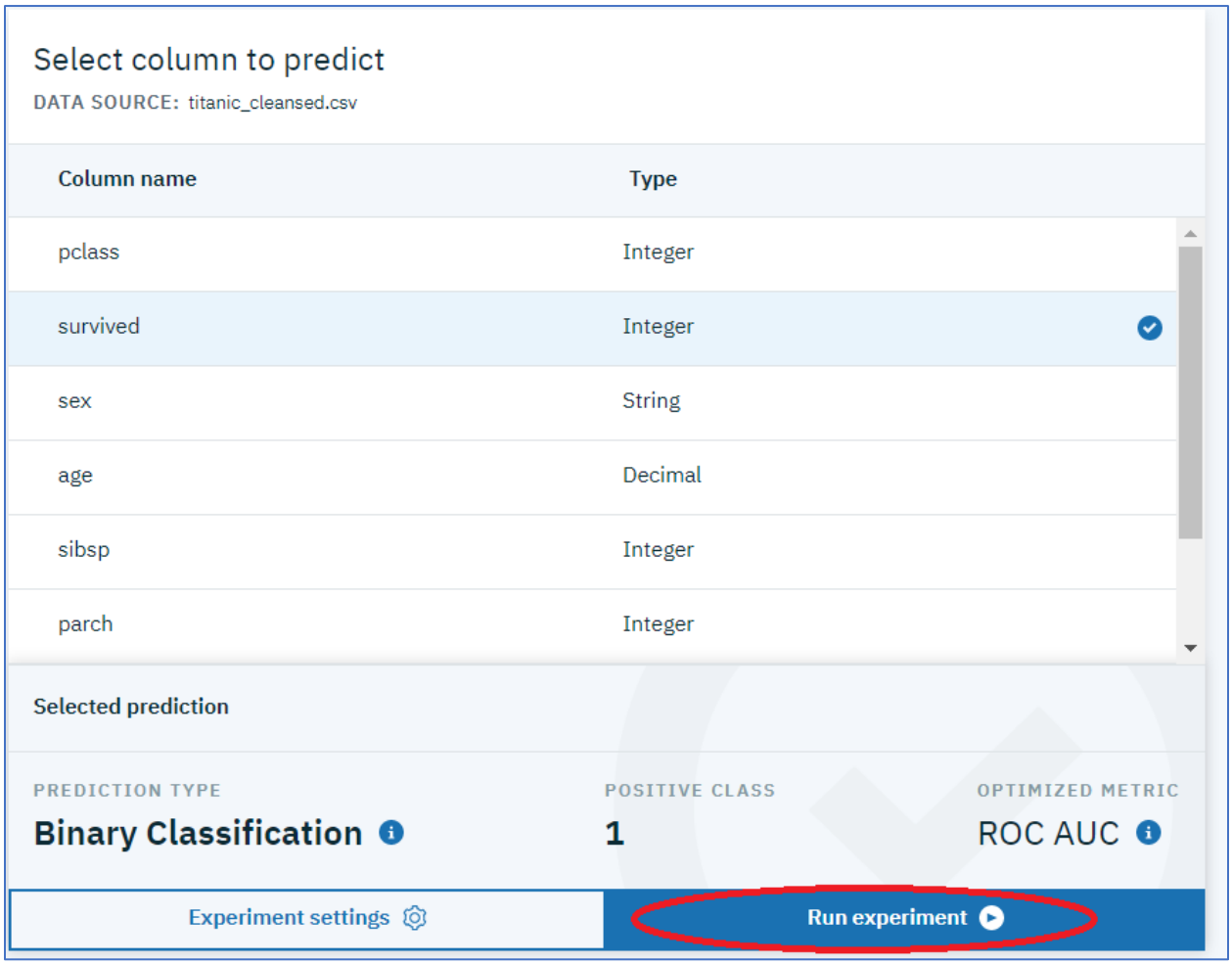
8. Click on **Prediction**.



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

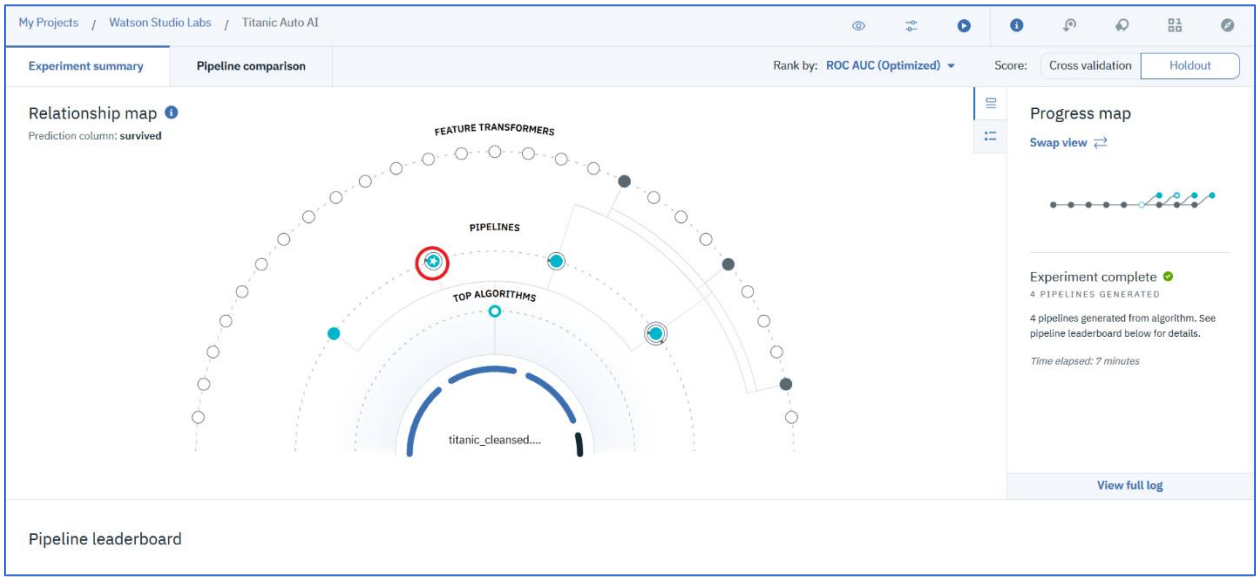


10. Click on **Run experiment**.



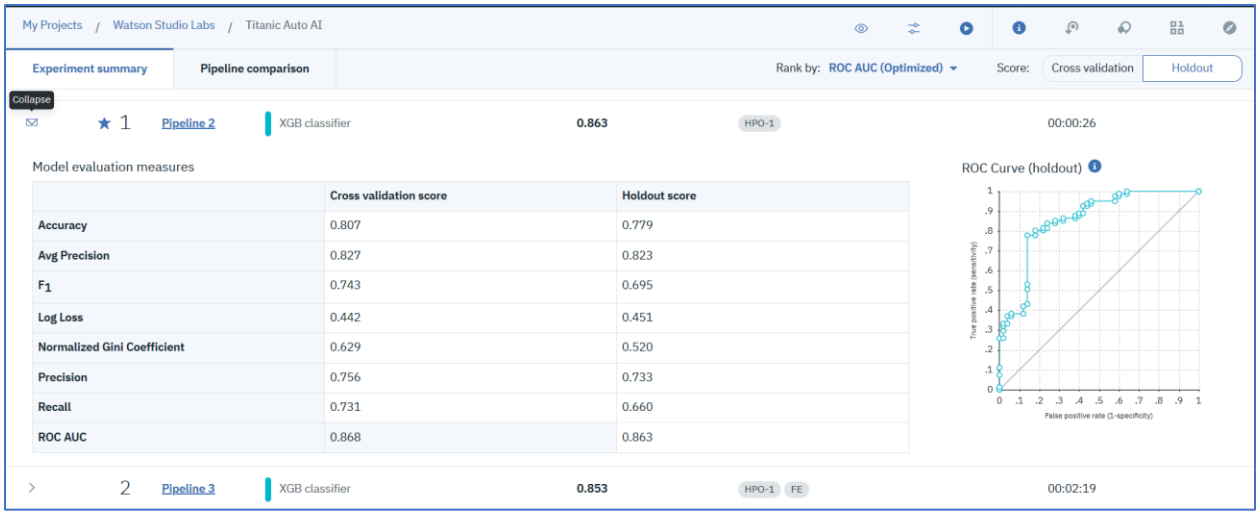
11. It will take several minutes for the eight alternative pipelines to be analyzed. The first pipeline picks the best algorithm. The second pipeline optimizes the hyper-parameters for the selected algorithm. The third pipeline does a feature transformation to try to improve the performance of the algorithm. 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 3 of the 8 pipelines have been completed. We are going to proceed as if Pipeline 2 is the best ranked pipeline. Yours could be different. Click on **Holdout**.

12. The best pipeline based on the performance metric (ROC AUC) is pipeline 2 for this run.

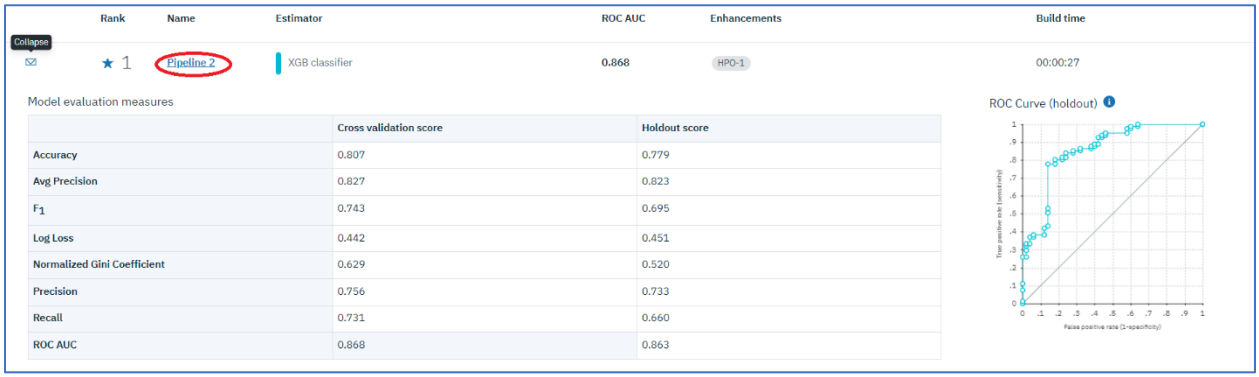


13. Scroll down to view the **Pipeline leaderboard**. The pipeline summary is then displayed. Click on the right arrow ▷ next to the top ranked pipeline.

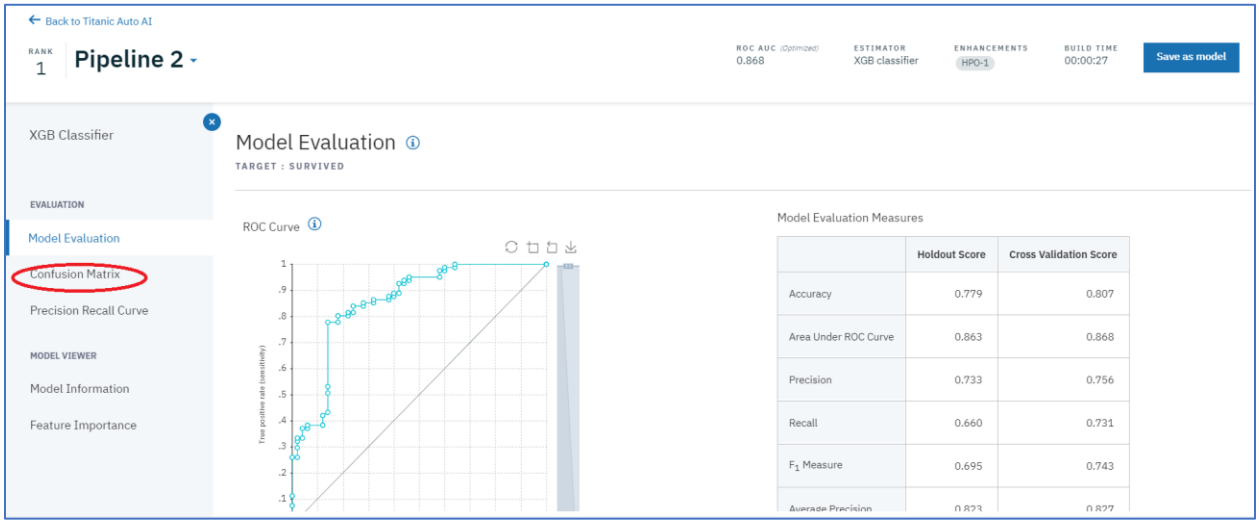
14. Scores are displayed for different metrics for both the training sample and the holdout sample.



15. Click on the **Pipeline** link for the top ranked pipeline.



16. The model evaluation metrics for the training and holdout sample are repeated. On the left are options for additional information. Click on the **Confusion Matrix**.

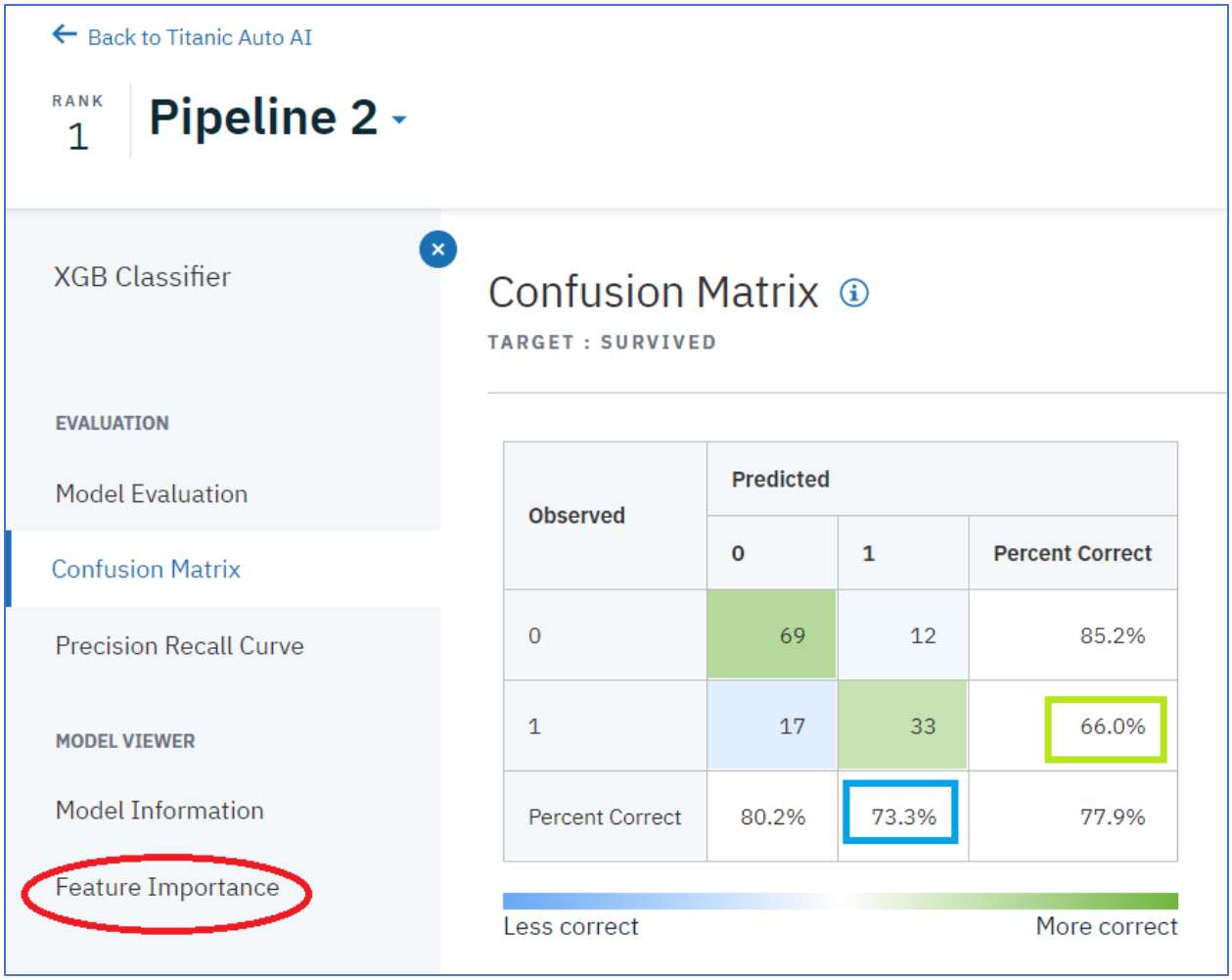


17. The Confusion Matrix is displayed for the holdout sample. The different metrics are computed based on the numbers in the Confusion Matrix. For example, Precision is defined by the percentage of predicted positives that are actually positive (i.e. the percentage of predicted survivors that survived).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

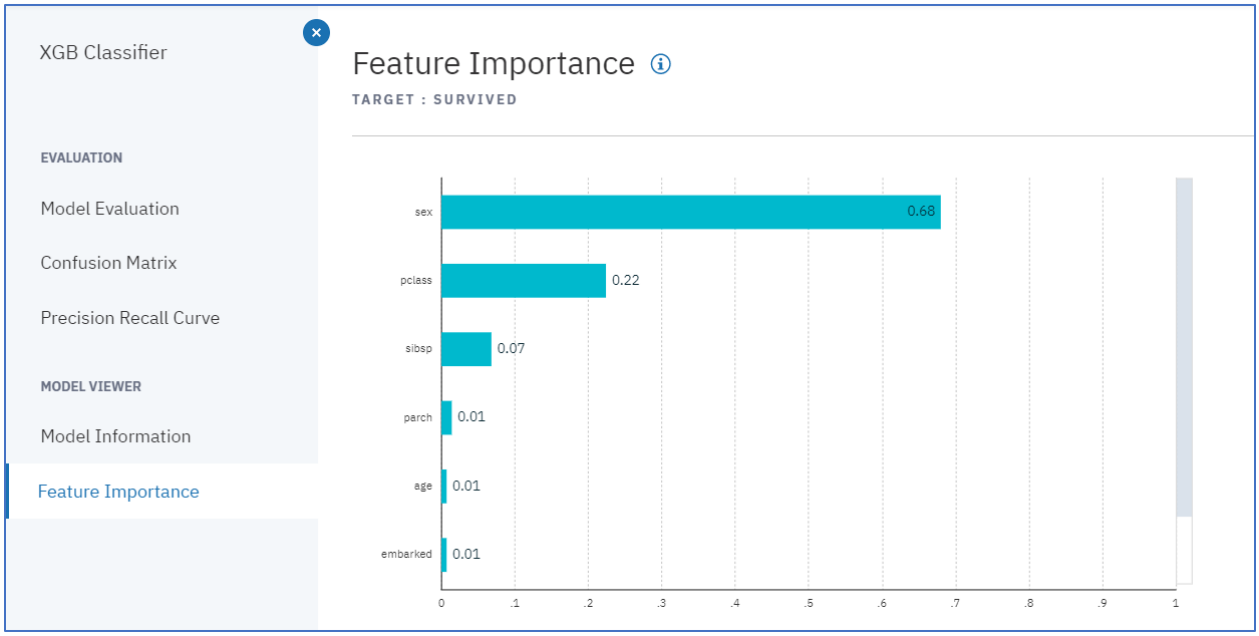
    Precision = True Positive/ (True Positive + False Positive) – shown inside blue rectangle on diagram below.

    Recall = True Positive/(True Positive + False Negative) - shown inside green rectangle on diagram below

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

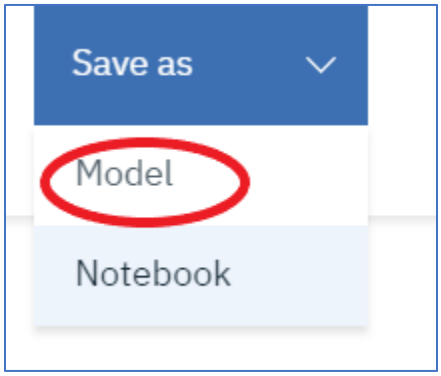


18. According to the Feature Importance, the sex variable is considered the most important feature followed by the passenger class.

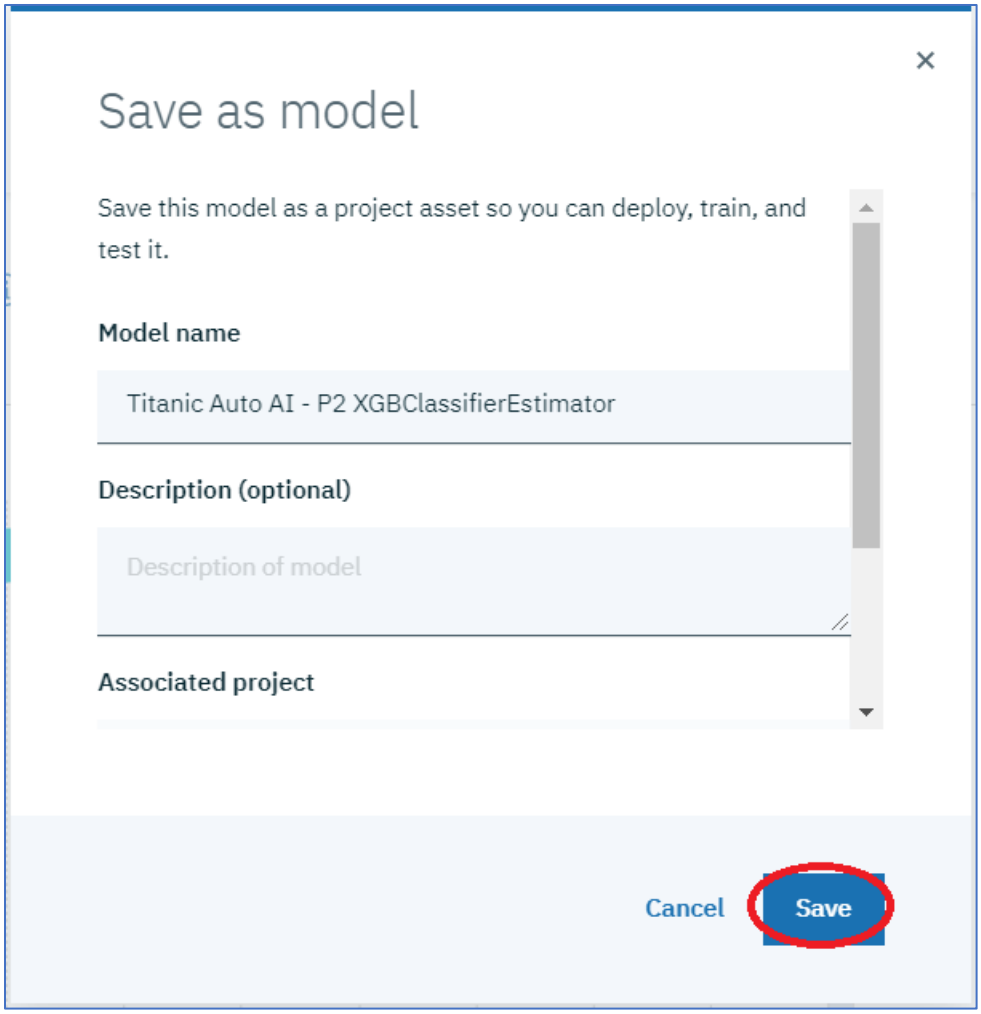


Step 3 – Save and Deploy the Selected Model

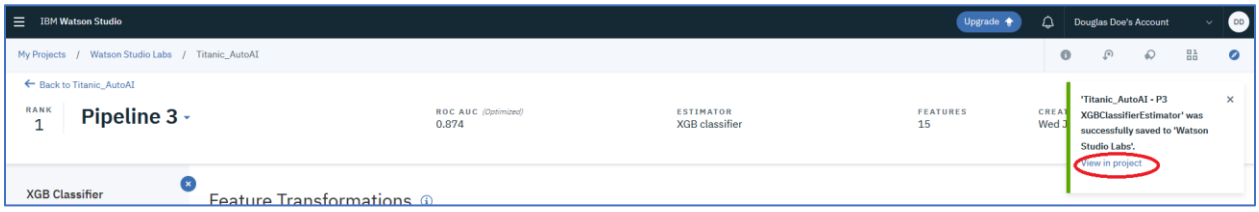
1. Click on **Save as**. Click on **Model**. Note you also have the option to save a Notebook. The notebook contains the code used to generate the pipeline. In this way a data scientist could use this as a starting point to tune the model even further.



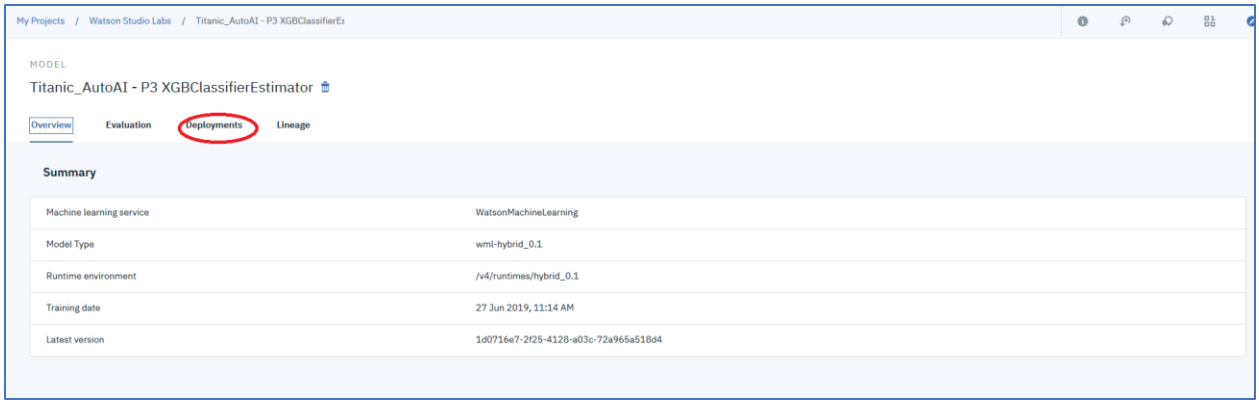
2. Optionally change the default name and click on **Save**.



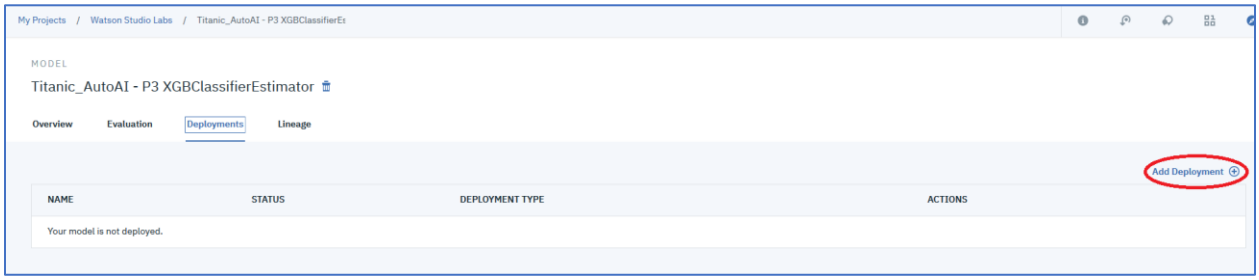
3. Click on **View in Project**.



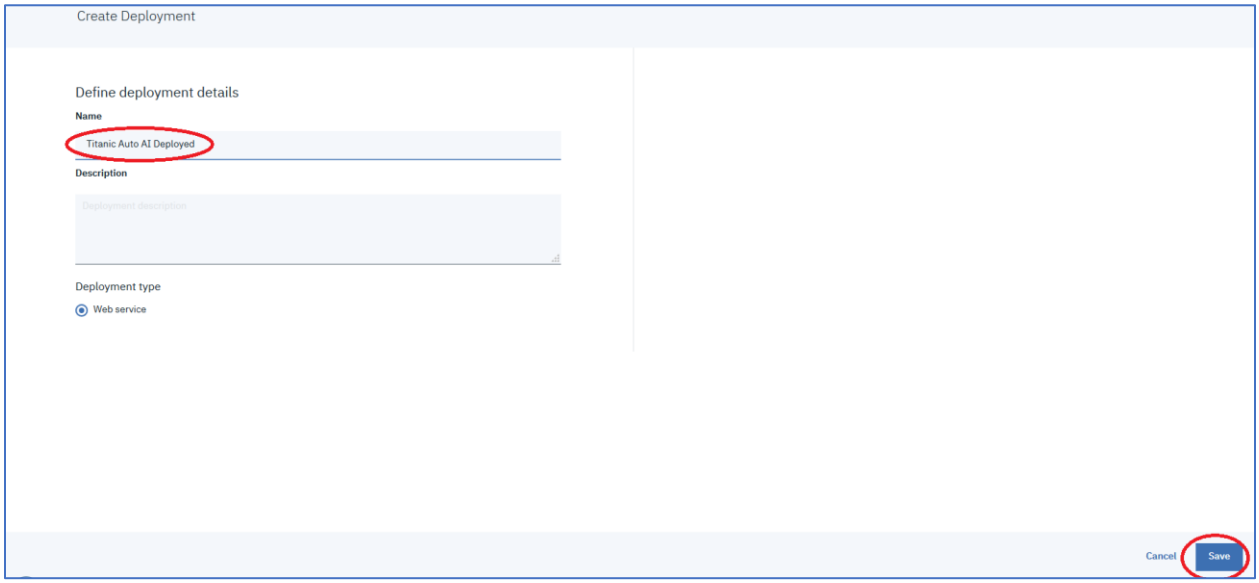
4. Click on **Deployments**



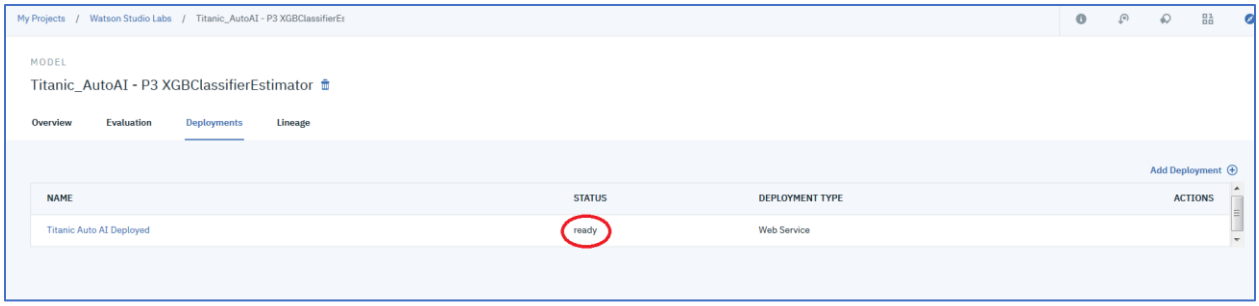
5. Click on **Add Deployment**.



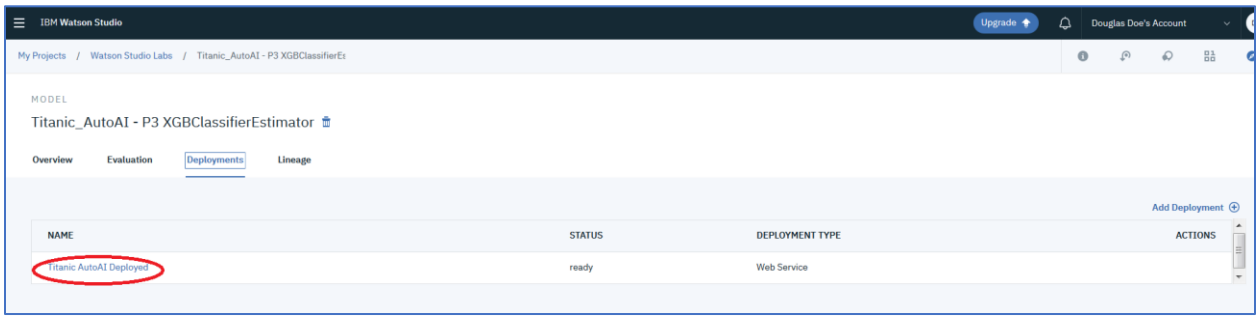
6. Enter a **Name** and click **Save**. Note, if status stays on **initializing** for a while, click on the browser refresh.



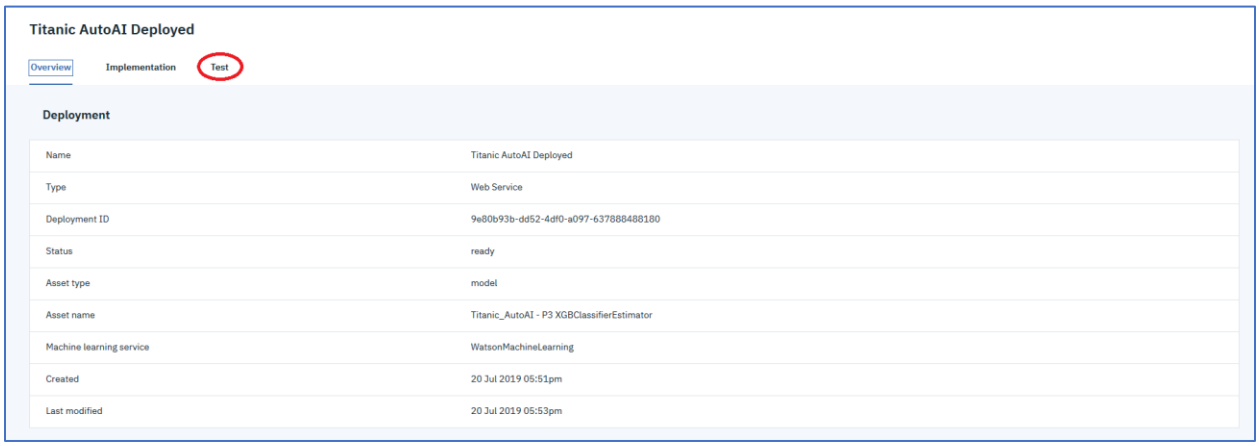
7. The model is successfully deployed on the IBM Cloud.



8. Click on **Titanic AutoAI Deployed**.

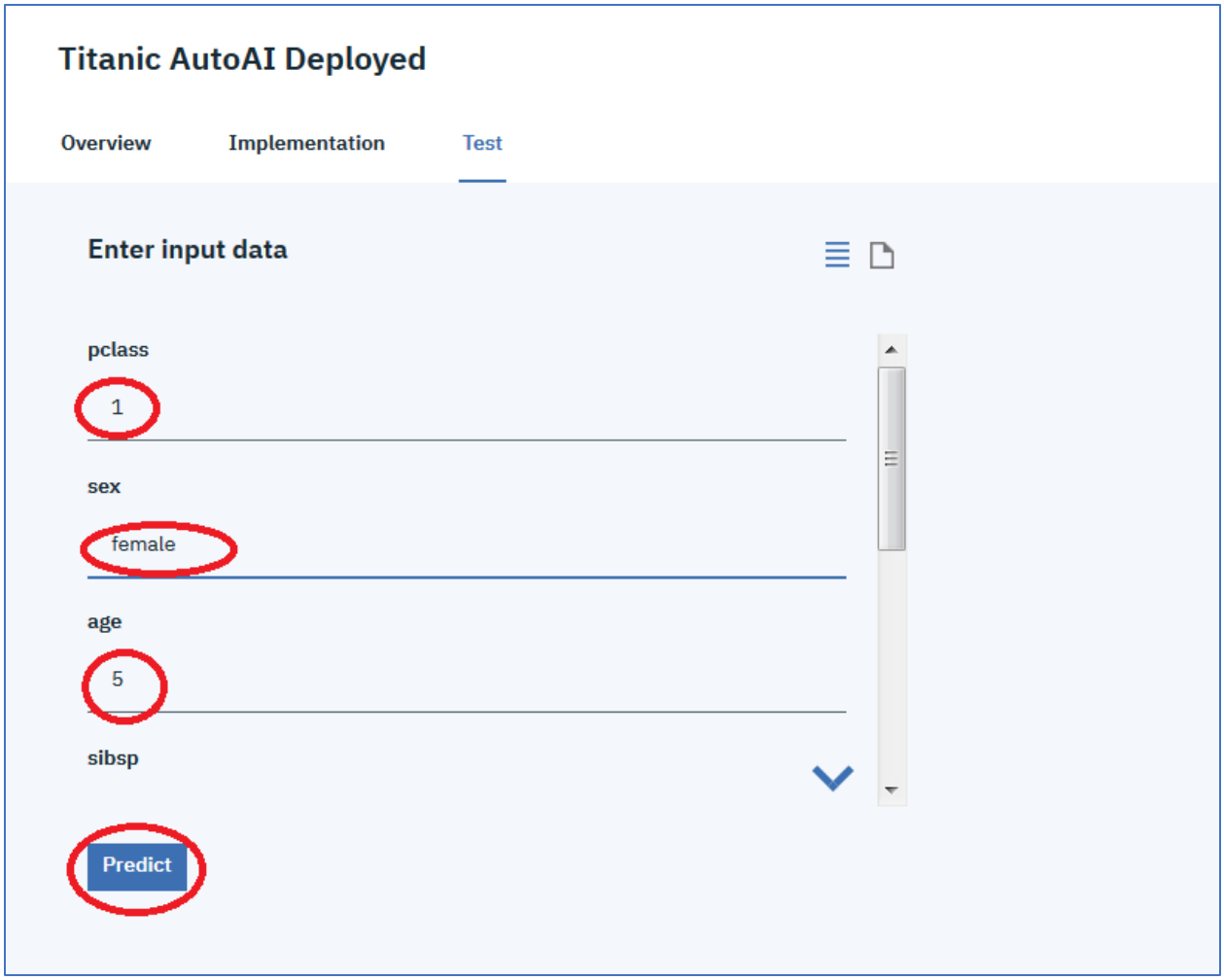


9. Click on **Test**.

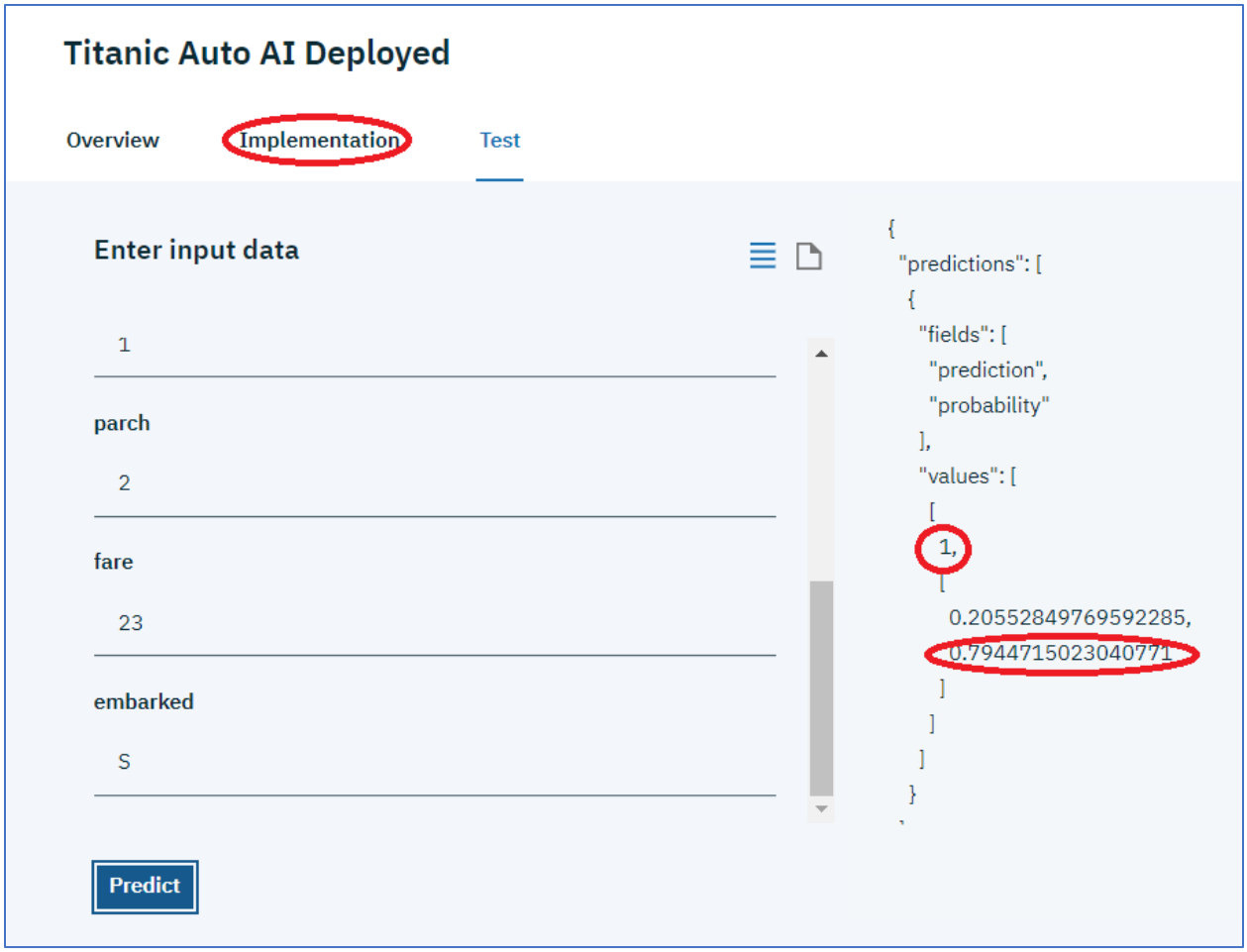


10. Enter values for a passenger. For example,
    pclass – 1
    sex – female
    age – 5
    sibsp – 1
    parch – 2
    fare – 23
    embarked – S

    and click **Predict.**



11. The model predicts this passenger would survive, with almost 80% confidence. Click on **Implementation**.



12. 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.