

Would you have survived the titanic? Try this step by step Machine Learning experiment to find out!

Rate this article ★★★★★



Susan Ibach (<https://social.msdn.microsoft.com/profile/Susan+Ibach>) May 13, 2016

0 (<https://blogs.msdn.microsoft.com/cdndevs/2016/05/13/would-you-have-survived-the-titanic-try-this-step-by-step-machine-learning-experiment-to-find-out/#respond>)

 Share 8

0

0



49:46

Tech Talk

- Apurva Thanky [MSFT]
(<https://social.msdn.microsoft.com/profile/Apurva+Thanky+%5BMSFT%5D>)
on UWP: New Controls (part 3 – Maps)
(<https://blogs.msdn.microsoft.com/cdndevs/2015/07/20/uwp-new-controls-part-3-maps/#comment-34805>)
- Sergiy Baydachnyy
(<https://social.msdn.microsoft.com/profile/Sergiy+Baydachnyy>)
on How to use SoftwareBitmap to display video from EZ-B controller
(<https://blogs.msdn.microsoft.com/cdndevs/2015/06/05/how-to-use-softwarebitmap-to-display-video-from-ez-b-controller/#comment-21285>)
- kishan kuman
(<http://sdit>) on Intel just open sourced Stephen Hawking's speech system and it's

the titanic!

If you prefer to learn with a video, check out this great video by Jennifer Marsman (<https://channel9.msdn.com/Blogs/raw-tech/Using-Azure-Machine-Learning-to-Predict-Who-Will-Survive-the-Titanic-Part-1>) for a more comprehensive introduction to data science and Azure Machine Learning Studio check out Data Science and Machine Learning Essentials on MVA (<https://mva.microsoft.com/en-US/training-courses/data-science-and-machine-learning-essentials-14100>)

Creating a Machine Learning Workspace

To use Azure Machine Learning Studio from your Azure account, you need a Machine Learning workspace. This workspace contains the tools you need to create, manage, and publish machine learning experiments.

To create a workspace, sign-in to your Microsoft Azure account.

How do I get access to Azure Machine Learning Studio?

Azure Machine Learning studio is part of Microsoft Azure. Microsoft Azure is a paid service, but there are a number of programs or trials you can use to explore it's capabilities

- You can try the guest sign in for Azure ML Studio (<https://studio.azureml.net/>) to explore many (but not all) features of Azure Machine Learning Studio
- Sign up for a one month free trial of Microsoft Azure (<https://azure.microsoft.com/en-us/pricing/free-trial/>)
- Students get some Azure features for free through DreamSpark, learn how to sign up for student Azure (<https://channel9.msdn.com/Series/Free-Cloud-for-Students>) (unfortunately Azure ML studio is not included in the DreamSpark/Azure offer)
- Will you use Machine Learning in your start-up? Check out Microsoft BizSpark (<https://www.microsoft.com/bizspark/>), a program for startups which includes Azure benefits through MSDN
- If you work at a company, ask if you have an MSDN subscription (<https://www.microsoft.com/bizspark/>), you may already have access to Azure
- If you want to use it in a course at school, faculty can apply for Azure Education grants (<https://www.microsoftazurepass.com/azureu>), which provide all students in the class with a 6 month Azure pass.
- If you want to use it for academic research, you can apply for Azure Research grants at [research.microsoft.com](http://research.microsoft.com/en-US/projects/azure/awards.aspx) (<http://research.microsoft.com/en-US/projects/azure/awards.aspx>)

1. Navigate to the Microsoft Azure portal portal.azure.com (<http://portal.azure.com/>) and log in using your Microsoft account credentials

2. In the Microsoft Azure portal create a new machine learning workspace. Select **+ New | Data + Analytics | Machine Learning**

a .NET 4.5 WinForms app that you can try for yourself (<https://blogs.msdn.microsoft.com/cdndevs/2015/08/14/intel-just-open-sourced-stephen-hawkings-speech-system-and-its-a-net-4-5-winforms-app-that-you-can-try-for-yourself/#comment-20535>)

- Kumar on Visual Studio Code and local web server (<https://blogs.msdn.microsoft.com/cdndevs/2016/01/24/visual-studio-code-and-local-web-server/#comment-19185>)
- dotMorten (<http://www.sharpgis.net>) on How to manage AllJoyn devices in C#: LIFX Color 1000 Example (<https://blogs.msdn.microsoft.com/cdndevs/2016/06/06/how-to-manage-alljoyn-devices-in-c-lifx-color-1000-example/#comment-19025>)

Developer Events



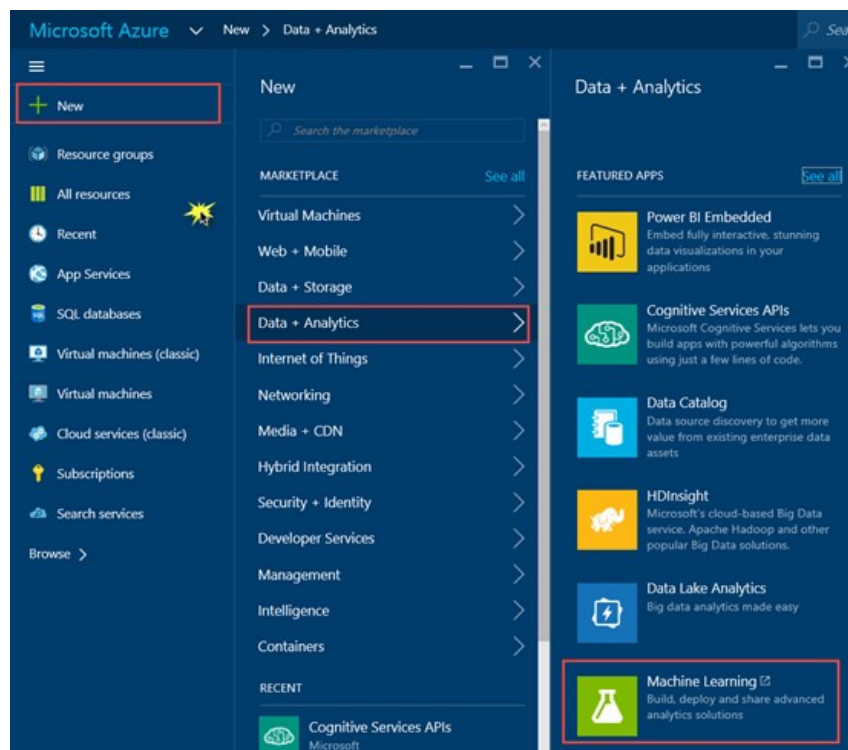
Microsoft De...

Like Page

Events

Timeline

Microsoft Developer does not have any upcoming events.



(<https://msdnshared.blob.core.windows.net/media/2016/05/image463.png>)

You will be redirected to the original Azure portal to enter the details for your machine learning workspace.

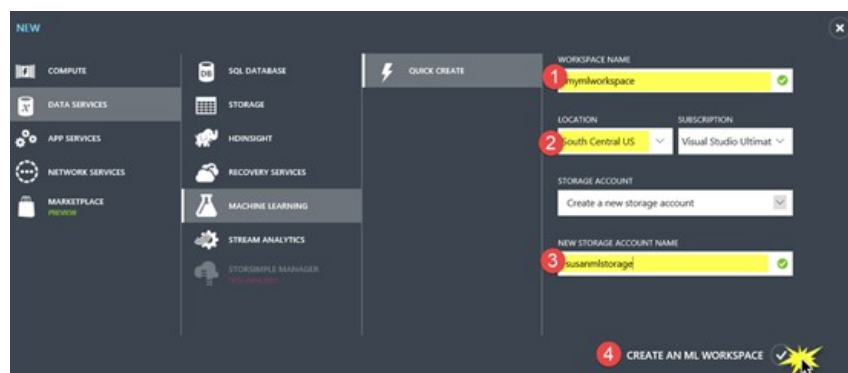
1. Enter a **WORKSPACE NAME** for your workspace

NOTE: Later, you can share the experiments you're working on by inviting others to your workspace. You can do this in Machine Learning Studio on the **SETTINGS** page. You just need the Microsoft account or organizational account for each user.

2. Specify the Azure **LOCATION**

3. Select an existing Azure **STORAGE ACCOUNT** or select **Create a new storage account** to create a new one and give your new storage account a name.

4. Select **CREATE AN ML WORKSPACE**.



(<https://msdnshared.blob.core.windows.net/media/2016/05/image464.png>)

Creating a new experiment in Azure Machine Learning Studio

After your Machine Learning workspace is created, you will see it listed on the portal under **MACHINE LEARNING**. At the time this post was created Machine Learning Workspaces are always displayed in the Azure Classic portal (even if you select the menu option from the new portal to create it), at some point the new portal will be updated so you can list them without going to the Classic view.

Follow us



(<https://twitter.com/cdndevs>)



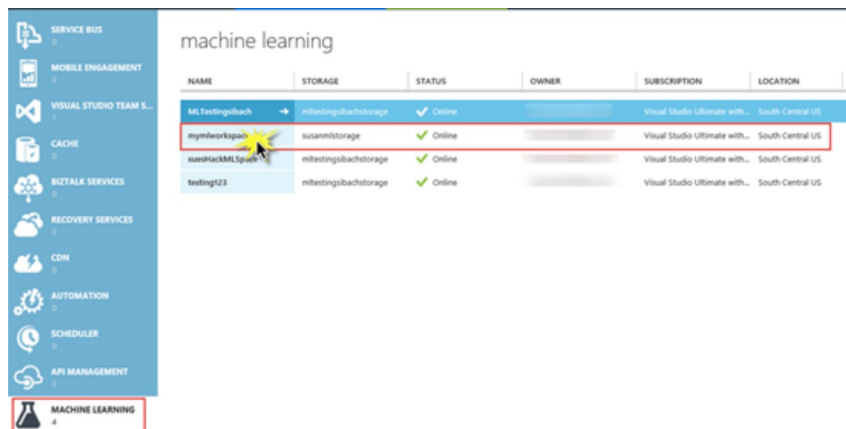
(<https://www.facebook.com/MicrosoftDeveloper.Canada>)



(<https://blogs.msdn.microsoft.com/cdndevs/feed/>)

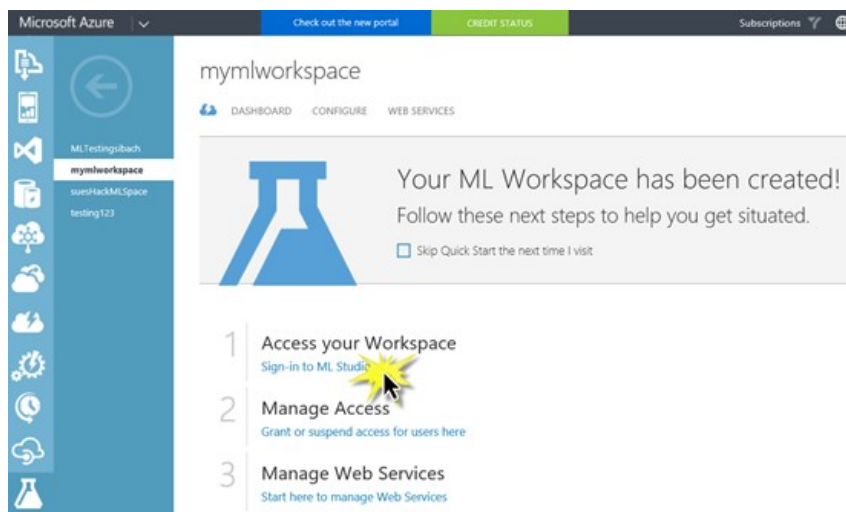


(<https://github.com/microsoft/>)



(<https://msdnshared.blob.core.windows.net/media/2016/05/SNAGHTML1c7641.png>)

Once your Machine Learning workspace is created, select your workspace from the list and then select Sign-in to ML Studio to access the Machine Learning Studio so you can create your first experiment!



(<https://msdnshared.blob.core.windows.net/media/2016/05/image465.png>)

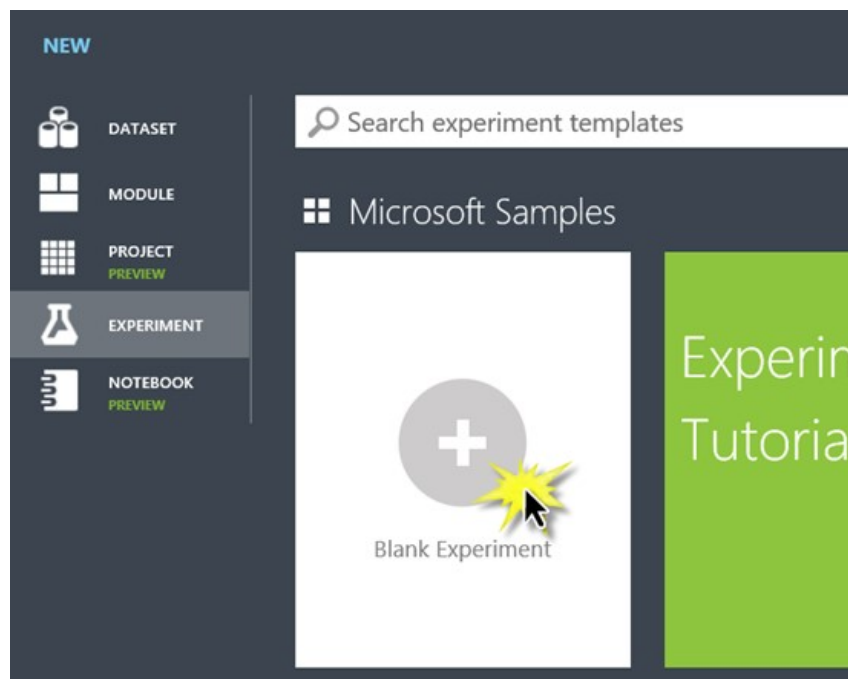
When prompted to take a tour select **Not Now**. You may want to take a tour later when you are exploring this tool on your own.

At the bottom of the screen select **+NEW**



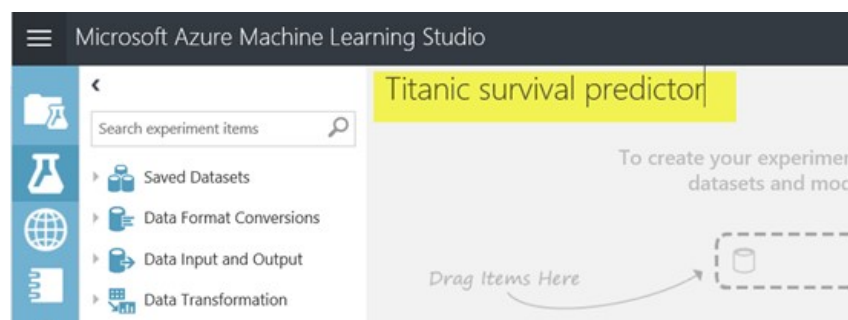
(<https://msdnshared.blob.core.windows.net/media/2016/05/image466.png>)

then select **+Blank Experiment**



(<https://msdnshared.blob.core.windows.net/media/2016/05/image467.png>)

Change the title at the top of the experiment to read **"Titanic survival predictor"**



(<https://msdnshared.blob.core.windows.net/media/2016/05/image468.png>)

Loading the data set

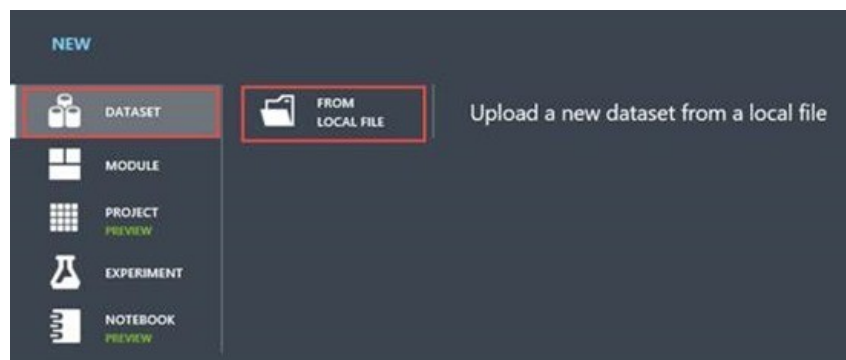
The titanic data set is not a sample data set already loaded in Azure Machine Learning Studio. It is an open data set you can download from various sources on the internet. Different files have slightly different columns and formats. The file used in this example is the train.csv Titanic file from Kaggle (<https://www.kaggle.com/c/titanic/data>) (this requires a Kaggle account, but if you are exploring machine learning, you might want to consider creating a Kaggle account to access other interesting datasets or to try some of their competitions).

I have renamed the train.csv file on my computer to TitanicSurvival.csv

Once you have downloaded the file you will need to create a dataset in Azure Machine Learning Studio for the titanic csv file.

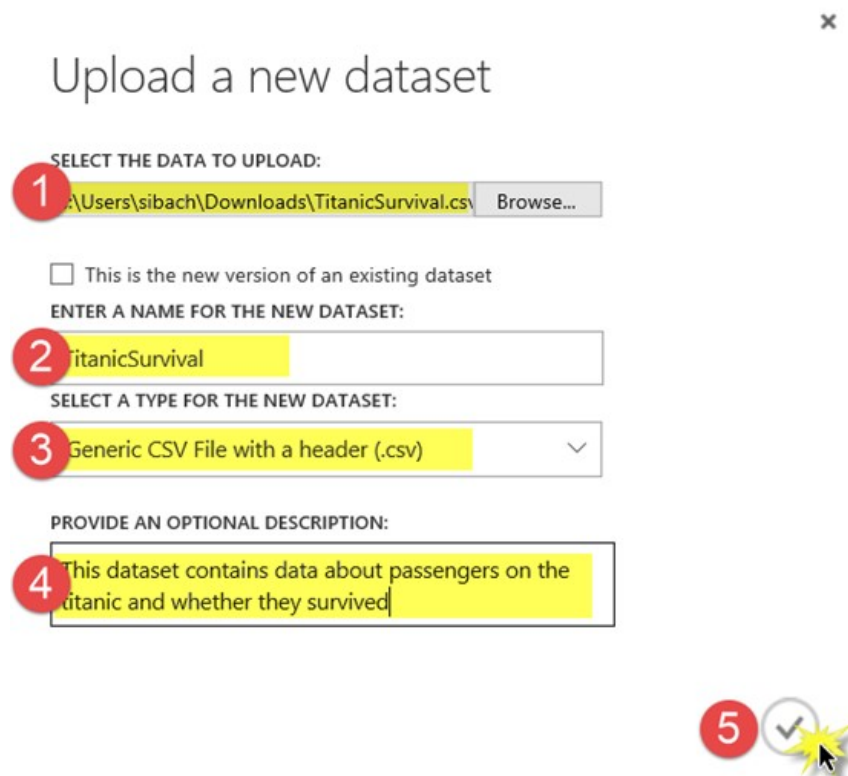
Select **+ NEW** at the bottom of the screen

Select **DATASET | FROM LOCAL FILE**



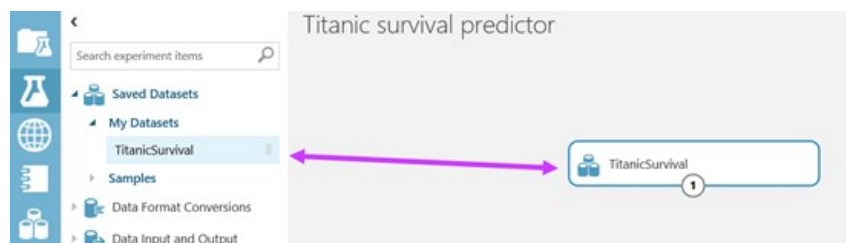
(https://msdnshared.blob.core.windows.net/media/2016/05/clip_image00262.jpg)

1. Select the **DATA TO UPLOAD** by browsing to select the csv file you downloaded containing the titanic data.
2. Enter **NAME FOR THE NEW DATASET**
3. Specify the **TYPE FOR THE NEW DATASET** as Generic CSV File with a header (.csv) this indicates we have a csv file and the first row of the csv file contains the headers for the data columns
4. Enter a description of the dataset to help you remember the dataset contents
5. Select the checkmark to start uploading the data into a dataset



(<https://msdnshared.blob.core.windows.net/media/2016/05/image469.png>)

Expand **Saved Datasets | My Datasets** and drag your newly created Titanic dataset to the experiment



(<https://msdnshared.blob.core.windows.net/media/2016/05/image470.png>)

Right click on the dataset on your worksheet and select **dataset | visualize** from the pop-up menu, explore the dataset by clicking on different columns. It's essential in Machine Learning to be familiar with

your data. This dataset contains information about passengers on the titanic and whether or not they survived.

Titanic survival predictor > TitanicSurvival > dataset

rows	columns										
891	12										
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	
view as											
	1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25	
	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.21	
	3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.92	
	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	

(<https://msdnshared.blob.core.windows.net/media/2016/05/image471.png>)

- **PassengerId** is a unique identifier assigned to each passenger
- **Survived** is a flag that indicates if a passenger survived. 0 = No, 1 = Yes
- **Pclass** is the passenger class. 1 = 1st class, 2 = 2nd class, 3 = 3rd class)
- **Name** is the name of the passenger
- **Sex** indicates the gender of the passenger
- **Age** indicates the age of the passenger
- **SibSp** indicates the number of siblings or spouses aboard the titanic with the passenger
- **Parch** indicates the number of parents or children aboard
- **Ticket** indicates the ticket number issued to the passenger
- **Fare** indicates the amount of money spent on their ticket
- **Cabin** indicates the cabin occupied by the passenger
- **Embarked** indicates the port where the passenger embarked. C = Cherbourg, Q = Queenstown, S = Southampton)

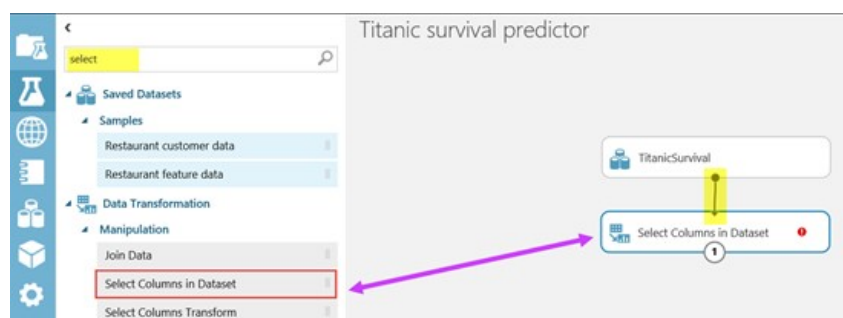
We are going to use Machine Learning to create a model that predicts whether fictional passengers (you? Your friends?) would have survived the sinking of the titanic.

Selecting Features for the Machine Learning Experiment

Some of the columns in the dataset are not meaningful for predicting whether or not a passenger would have survived. We know they evacuated women and children first, so age and gender are certainly important columns (referred to as features) that we want to consider when making our predictions but PassengerId is just a number assigned to each passenger, and the name of the passenger isn't going to help us predict survival either.

Let's select only the significant features in our dataset to use in our machine learning experiment.

Type "**Select**" into the search bar and drag the **Select Columns in Dataset** task to the workspace. Connect the output of your dataset to the project columns task input



(<https://msdnshared.blob.core.windows.net/media/2016/05/image472.png>)

The **Select Columns in Dataset** task allows you to specify which columns in the data set you think are significant to a prediction (i.e. your features). You need to look at the data in the dataset and decide which columns represent data that you think will affect whether or not a passenger would survive. You also need

to select the column you want to predict. In this case we are going to try to predict the value of Survived. This is a 0/1 column that indicates whether a passenger survived the sinking of the titanic.

Click on the **Select Columns in Dataset** task. On the properties pane on the right hand side, select **Launch column selector**

Properties Project

Select Columns in Dataset

Select columns

Selected columns:
Launch the selector tool
to make a selection

Launch column selector

(<https://msdnshared.blob.core.windows.net/media/2016/05>

/image473.png) Select the columns you think affect whether or not a passenger would have survived as well as the column we want to predict: **Survived**. In the following screenshot, I selected **Survived, Pclass, Sex, Age, Parch, and SibSp**.

Select columns



(<https://msdnshared.blob.core.windows.net/media/2016/05/image474.png>)

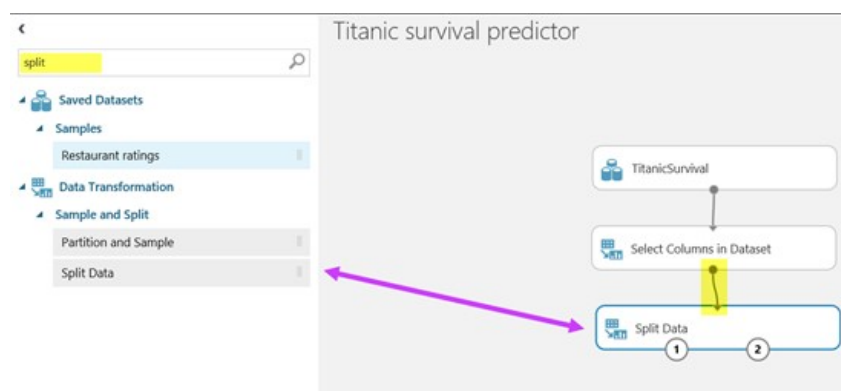
Setting aside data for testing

Whenever we execute machine learning experiments, we use some of our data to train the model and we put some data aside to test the model. In Azure Machine Learning Studio, we use the **Split Data** task to put aside data for testing.

The Split Data task allows us to divide up our data, we need some of the data to try and find patterns and we need to save some of the data to test if the model we create successfully makes predictions.

Traditionally you will split the data 80/20 or 70/30.

Type **"split"** into the search bar and drag the **Split Data** task to the workspace. Connect the output of Project Columns task to the input of the Split Data task.



(<https://msdnshared.blob.core.windows.net/media/2016/05/image475.png>)

Click on the Split Data task to bring up properties, specify **.8** as the **Fraction of rows in the first output**

Properties Project

Split Data

Splitting mode

Split Rows

(<https://msdnshared.blob.core.windows.net/media/2016/05>

Fraction of rows in the fir...

0.8

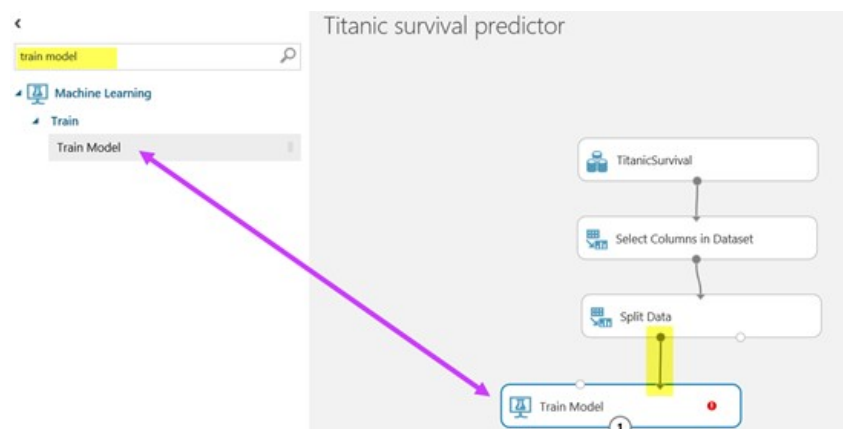
☒ Randomized split

/image476.png)

Training the model

Now we can get Azure Machine Learning Studio to train the model so we can find the patterns in the historical data to make predictions for new records.

Type **"train model"** into the search bar. Drag the **train model task** to the workspace. Connect the first output (the one on the left) of the Split Data task to the rightmost input of the Train model task. This will take 80 % of our data and use it to train/teach our model to make predictions.



(<https://msdnshared.blob.core.windows.net/media/2016/05/image477.png>)

We need to tell the train model task which column we are trying to predict with our model. In our case we are trying to predict the value of the column **Survived** which indicates if a passenger survived the sinking of the titanic.

Click on the **Train Model** task. In the properties window select **Launch Column Selector**. Select the column **Survived**.

Select a single column

BY NAME
WITH RULES

Include column names Survived X

(<https://msdnshared.blob.core.windows.net/media/2016/05/image478.png>)

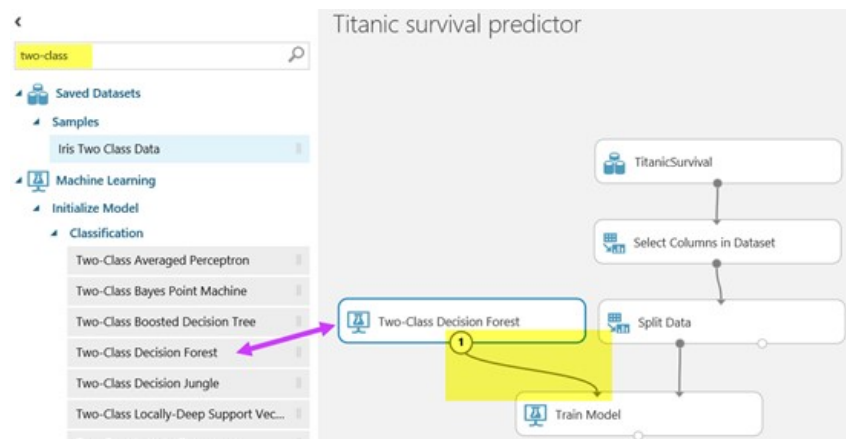
If you are a data scientist who creates your own algorithms, you could now import your own R code to try and analyze the patterns. But, we can also use one of the existing built-in standard algorithms.

Different types of machine learning, use different algorithms. Since we are trying to predict if an output has one of two values we want to use a two-class algorithm to train our model. Two-clas algorithms are used to predict outcomes that can only have two possible values. In our case a value of 1 or 0 which

indicates survival.

Type **"two-class"** into the search bar. You will see a number of different classification algorithms listed. Each algorithm has its advantages and disadvantages. Check out the Azure Machine Learning Studio Cheat Sheet (<https://azure.microsoft.com/en-us/documentation/articles/machine-learning-algorithm-cheat-sheet/>) for a quick reference guide to algorithm selection. I am going to select the **Two-Class Decision Forest** to train my model. Select one of the two-class algorithms and drag it to the workspace.

Connect the output of the Algorithm task to the leftmost input of the train model task.

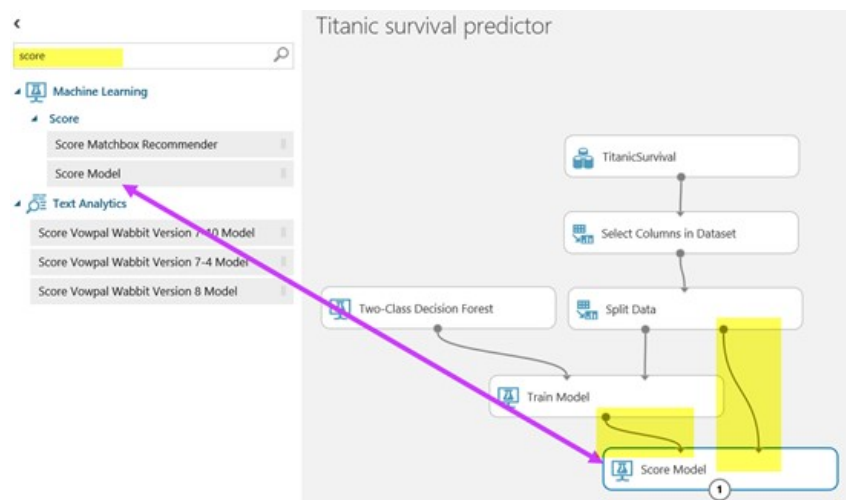


(<https://msdnshared.blob.core.windows.net/media/2016/05/image479.png>)

Testing your model

After the model is trained, we need to see how well it predicts survival, so we need to score the model by having it test against the 20% of the data we split to our second output using the Split Data task.

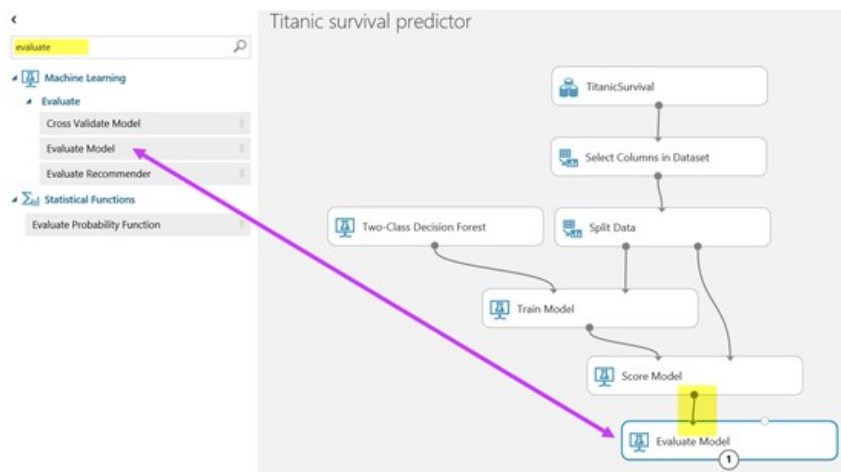
Type **"score"** into the search bar and drag the **Score Model** task to the workspace. Connect the output of Train Model to the left input of the Score model task. Connect the right output of the Split Data task to the right input of the Score Model task as shown in the following screenshot.



(<https://msdnshared.blob.core.windows.net/media/2016/05/image480.png>)

Now we need a report on our test results.

Type **"evaluate"** into the search bar and drag the **Evaluate Model** task to the bottom of the workspace. Connect the output of the Score model task to the left input of the Evaluate Model task.



(<https://msdnshared.blob.core.windows.net/media/2016/05/image481.png>)

You are now ready to run your experiment!

Press **Run** on the bottom toolbar. You will see green checkmarks appear on each task as it completes. When the entire experiment is completed you can check how well your model makes predictions.

How to interpret your results

To see your test results, right click on the evaluate model task and select "**Evaluation results | Visualize**".

The closer the graph is to a straight diagonal line the more your model is guessing randomly. You want your line to get as close to the upper left corner as possible.



(https://msdnshared.blob.core.windows.net/media/2016/05/clip_image002123.jpg)

If you scroll down you can see the detailed results. AUC (Area Under Curve) is a great overall indicator of your model performance. The closer AUC is to 1, the better your model is making predictions.

You can also see the number of false and true positive and negative predictions

- True positives are how often your model correctly predicted someone would survive
- False positives are how often your model predicted a flight would survive, when they did not survive (i.e. your model predicted incorrectly)
- True negatives indicate how often your model correctly predicted a passenger would not survive
- False negatives indicate how often your model predicted a flight would be not survive, when in fact they did survive (i.e. your model predicted incorrectly)

You want high values for True positives and True negatives, you want low values for False Positives and False negatives.

True Positive 48	False Negative 19	Accuracy 0.809	Precision 0.762	Threshold 0.5	AUC 0.826
False Positive 15	True Negative 96	Recall 0.716	F1 Score 0.738		
Positive Label 1	Negative Label 0				

(<https://msdnshared.blob.core.windows.net/media/2016/05/image482.png>)

Creating a web service for your trained model

Once you have trained a model with a satisfactory level of accuracy, how do you use it? One of the great things about Azure Machine Learning Studio is how easy it is to take your model and deploy it as a web service. Then you can simply have a website or app call the web service, pass in a set of values for the project columns and the web service will return the predicted value and confidence of the result.

Convert the training experiment to a predictive experiment

Once you've trained your model, you're ready to use it to make predictions for new data. To do this, you convert your training experiment into a predictive experiment. By converting to a predictive experiment, you're getting your trained model ready to be deployed as a web service. Users of the web service will send input data to your model and your model will send back the prediction results.

To convert your training experiment to a predictive experiment, click **Run** at the bottom of the experiment canvas, then select Set Up Web Service

Azure ML Deploying Web service demo

Creating a web service for your trained model

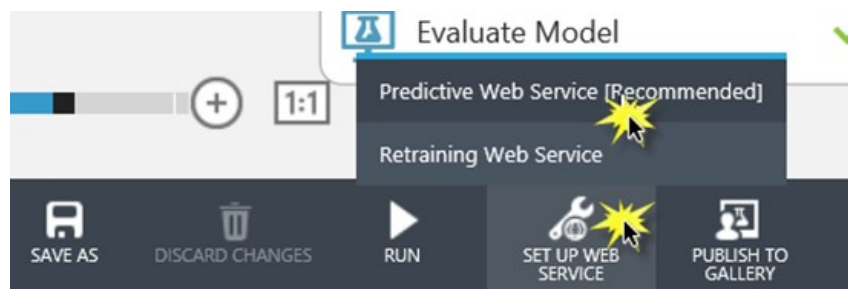
Once you have trained a model with a satisfactory level of accuracy, how do you use it? One of the great things about Azure Machine Learning Studio is how easy it is to take your model and deploy it as a web service. Then you can simply have a website or app call the web service, pass in a set of values for the project columns and the web service will return the predicted value and confidence of the result.

Convert the training experiment to a predictive experiment

Once you've trained your model, you're ready to use it to make predictions for new data. To do this, you convert your training experiment into a predictive experiment. By converting to a predictive experiment, you're getting your trained model ready to be deployed as a web service. Users of the web service will send input data to your model and your model will send back the prediction results.

To convert your training experiment to a predictive experiment, click **Run** at the bottom of the experiment canvas, then select Set Up Web Service

Select **Set Up Web Service**, then select **Predictive Web Service**.

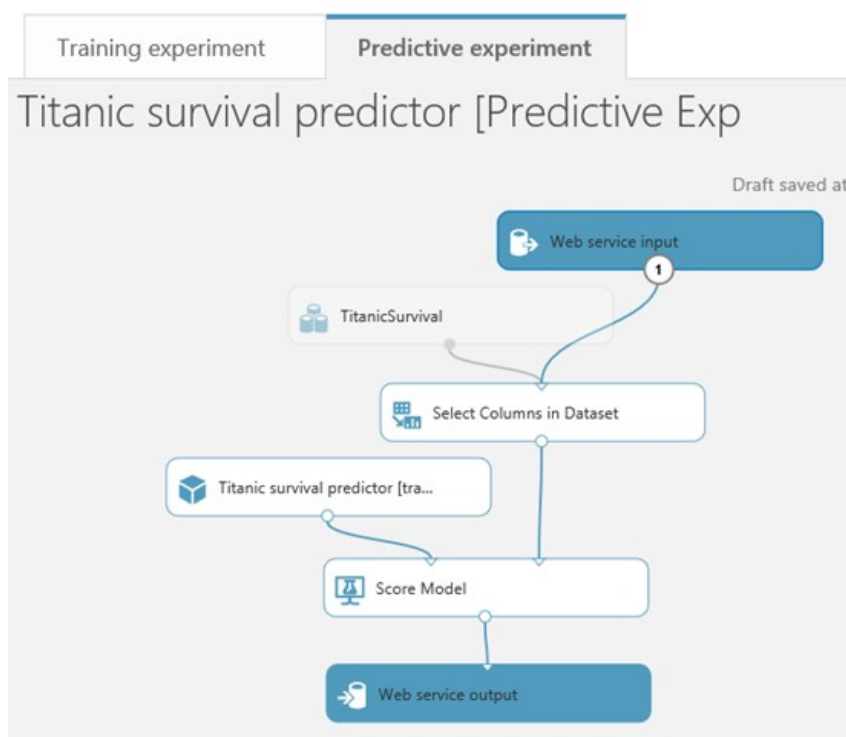


(https://msdnshared.blob.core.windows.net/media/2016/05/clip_image0017.png)

This will create a new predictive experiment for your web service. The predictive model doesn't have as many components as your original experiment, you will notice a few differences:

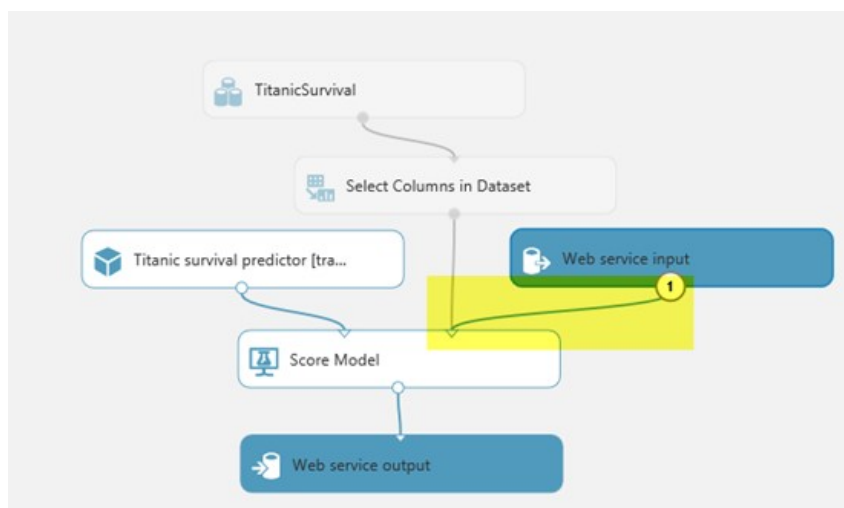
- You don't need the data set because when someone calls the web services they will pass in the data to use for the prediction.
- You still need to identify which columns will be used for predictions if you pass in a full record of data.
- Your algorithm and Train Model tasks have now become a single trained model which will be used to analyze the data passed in and make a prediction

- We don't need to evaluate the model to test it's accuracy. All we need is a Score model to return a result from our trained model.
- Two new tasks are added to indicate how the data from the web service is input to the experiment, and how the data from the experiment is returned to the web service.



(<https://msdnshared.blob.core.windows.net/media/2016/05/image483.png>)

Delete the connection from the Web input to **Select Columns in Dataset** task and redraw the connection from the Web input to the **Score Model** task. If you leave the web input connected to project columns, the web service will prompt you for values for all the data columns even though we don't use them to make our prediction. If you have the web input connected to the score model directly, the web service will only expect the data columns we selected in our **Select Columns in DataSet** task which we determined are relevant for making predictions.



(<https://msdnshared.blob.core.windows.net/media/2016/05/image484.png>)

For more details on how to do this conversion, see [Convert a Machine Learning training experiment to a predictive experiment](https://azure.microsoft.com/en-us/documentation/articles/machine-learning-convert-training-experiment-to-scoring-experiment/) (<https://azure.microsoft.com/en-us/documentation/articles/machine-learning-convert-training-experiment-to-scoring-experiment/>)

Deploy the predictive experiment as a web service

Now that the predictive experiment has been sufficiently prepared, you can deploy it as an Azure web

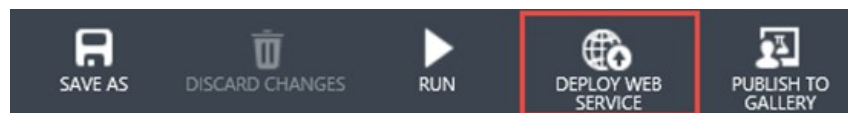
service. Using the web service, users can send data to your model and the model will return its predictions.

To deploy your predictive experiment,

click **Run** at the bottom of the experiment canvas

After it runs successfully

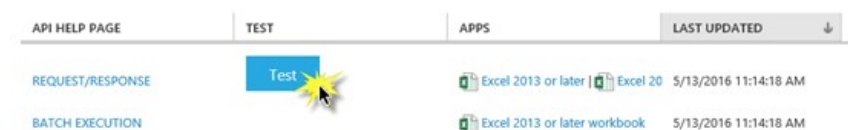
Select **Deploy Web Service**. The web service is set up and you are placed in the web service dashboard.



(<https://msdnshared.blob.core.windows.net/media/2016/05/image485.png>)

Test the web service

Select the **Test** link in the web service dashboard. A dialog pops up to ask you for the input data for the service. These are the columns expected by the scoring experiment. Enter a set of data and then select **OK**. The results generated by the web service are displayed at the bottom of the dashboard.



(<https://msdnshared.blob.core.windows.net/media/2016/05/image486.png>)

You may have to scroll down to see all the fields you need to enter

- Survived – leave as 0, that is the value the web service will predict
- PClass - Enter 1,2 or 3 for first, second or third class
- Sex – Enter male or female (lowercase letters)
- Age – Enter a numeric age
- SIBSP – Enter the number of siblings and spouses travelling with the passenger
- PARCH – Enter the number of parents and children travelling with the passenger

(<https://msdnshared.blob.core.windows.net/media/2016/05/image487.png>)

The results of the test will appear at the bottom of the screen.

Select **Details** to see the full record returned

You will see the record you entered followed by the predicted output and the probability (columns **scored label**, and **scored probabilities** respectively). In the screenshot below there is a .125 (12.5%) probability my imaginary passenger would have perished on the titanic (predited outcome for survived is 0). The

Sample Code

C# Python R

Select sample code

```
using (var client = new HttpClient())
{
    var scoreRequest = new
    {
        Inputs = new Dictionary<string, StringTable>() {
            {
                "inputs",
                new StringTable()
                {
                    ColumnNames = new string[] { "Survived", "Pclass", "Sex", "Age", "SibSp", "Parch" },
                    Values = new string[,] { { "0", "0", "value", "0", "0", "0" }, { "0", "0", "value", "0", "0", "0" }, }
                },
            },
        },
        GlobalParameters = new Dictionary<string, string>() {
        };
    };
    var apiKey = "abc123"; // Replace this with the API key for the web service
    client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", apiKey);

    client.BaseAddress = new Uri("https://usouthcentral.services.azureml.net/workspaces/d91fc87377044b06acd4c354de3944e9/services/8ef2128c-...");

    // WARNING: The 'await' statement below can result in a deadlock if you are calling this code from the UI thread of an ASP.NET applicat
    // One way to address this would be to call ConfigureAwait(false) so that the execution does not attempt to resume on the original cont
    // For instance, replace code such as:
    // ...
    // ...
}
```

(<https://msdnshared.blob.core.windows.net/media/2016/05/image491.png>)

Replace the apiKey of abc123 with the API key displayed in the dashboard of your web service.

Replace the values with the values you wish to pass into the web service and you can now call the web service from your code to retrieve predictions!

For more information about accessing a Machine Learning web service, see [How to consume a deployed Azure Machine Learning web service](https://azure.microsoft.com/en-us/documentation/articles/machine-learning-consume-web-services/) (<https://azure.microsoft.com/en-us/documentation/articles/machine-learning-consume-web-services/>).

Now get your friends to enter their data and find out who would survive the sinking of the titanic!

Tags [Azure](https://blogs.msdn.microsoft.com/cdn devs/tag/azure/) (<https://blogs.msdn.microsoft.com/cdn devs/tag/azure/>) [Azure Machine Learning Studio](https://blogs.msdn.microsoft.com/cdn devs/tag/azure-machine-learning-studio/) (<https://blogs.msdn.microsoft.com/cdn devs/tag/azure-machine-learning-studio/>) [How to](https://blogs.msdn.microsoft.com/cdn devs/tag/how-to/) (<https://blogs.msdn.microsoft.com/cdn devs/tag/how-to/>) [Machine Learning](https://blogs.msdn.microsoft.com/cdn devs/tag/machine-learning/) (<https://blogs.msdn.microsoft.com/cdn devs/tag/machine-learning/>) [Titanic](https://blogs.msdn.microsoft.com/cdn devs/tag/titanic/) (<https://blogs.msdn.microsoft.com/cdn devs/tag/titanic/>)

Comments (0)

You must be logged in (https://blogs.msdn.microsoft.com/cdn devs/wp-login.php?redirect_to=https%3A%2F%2Fblogs.msdn.microsoft.com%2Fcdn devs%2F2016%2F05%2F13%2Fwould-you-have-survived-the-titanic-try-this-step-by-step-machine-learning-experiment-to-find-out%2F) to post a comment.