**Lab for Chapter 7:**

**End to End Predictive Analytics Workflow**

Contents

[Exercise 1: Finalizing the Model **4**](#_bookmark0)

[Exercise 2: Setting up the Web Service **6**](#_bookmark1)

[Exercise 3: Deploying Web Service **9**](#_bookmark2)

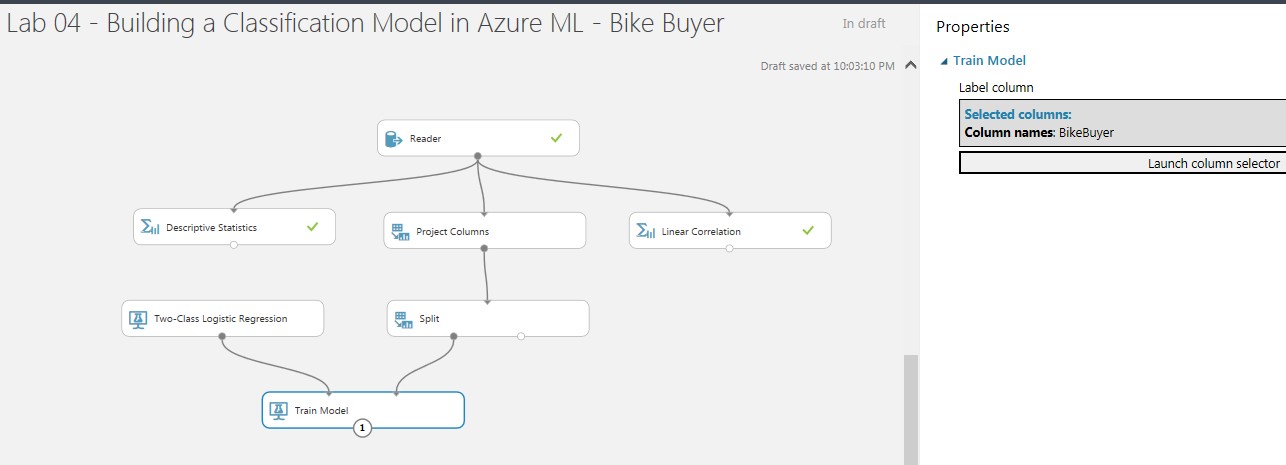
[Exercise 4: Testing your Model **11**](#_bookmark3)

**Exercise 0: Overview**

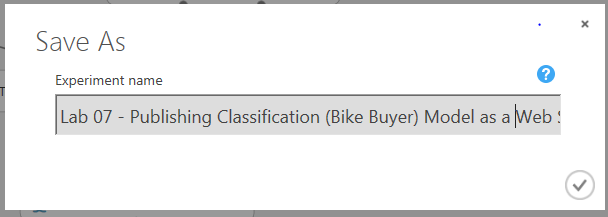
This exercise deals with how to set up a Web Service for your Predictive model and once your experiment is published as a Web Service, testing the API Request/Response by providing live data.

**Exercise 1: Finalizing the Model**

1. For this exercise, we will use the experiment built for evaluating the Classification Model (Lab for Chapter 4 – Logistic Regression (Bike Buyer))
2. Make sure you are happy with the model that you have trained and built using the best possible Algorithm. In Lab 04, the model is built using “**Two-Class Logistic Regression**”

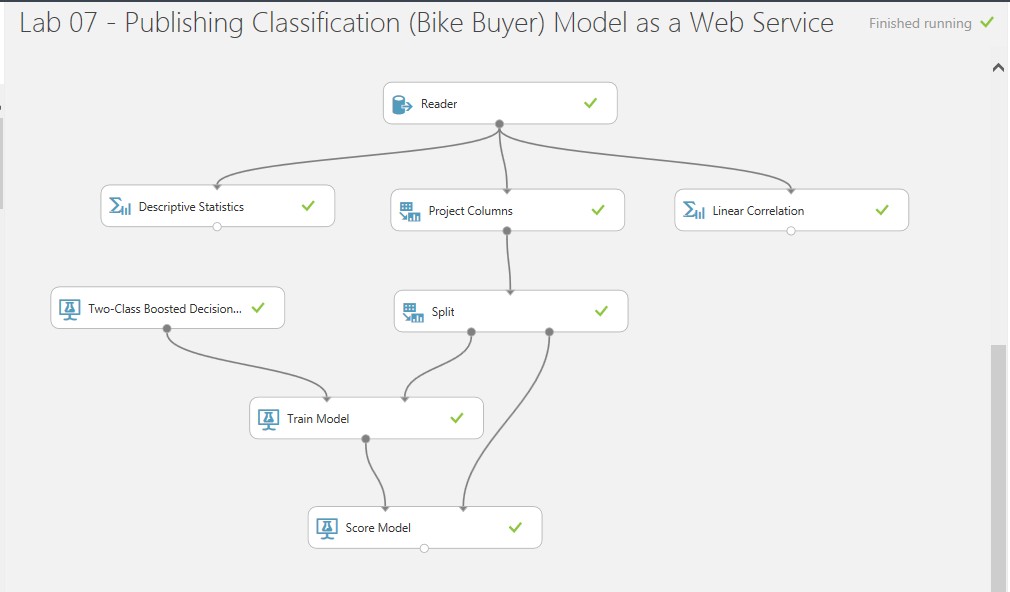


1. Create a new copy of your experiment by hitting “SAVE AS” on the bottom middle toolbar and rename it to “**Lab 07 - Publishing Classification (Bike Buyer) Model as a Web Service**”



**Rationale**: In the future if we wanted to go back and re-optimize our model, we still have our original experiment preserved.

1. We want to remove any unnecessary processing modules from our workspace’s experiment because once it is a web service, we do not want to be charged for unnecessary processing power and we want to speed up each individual iteration of a request call.
2. Remove the “Evaluate Model” module by selecting it then hitting the “Delete” key on your keyboard (if relevant)>
3. Run the experiment to ensure that all modules have a green check mark next to them with no errors.



Notice how all the modules have green check marks (please note that our example will have Two Class Logistic Regression instead of Two Class Boosted Decision Trees)

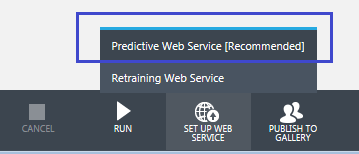
1. Click the output of the “Score Model” and select ‘Visualize’ to visualize the output of our Predictive Model.



* 1. You should see the ***Scored Probabilities***column and the ***Scored Labels***column.
  2. The Scored Probabilities is the prediction that the algorithm has made. (% chance of buying a bike).
  3. The Scored Label is set to “Yes” or “No” meaning “Bike Buyer” or “Not a Bike Buyer”.

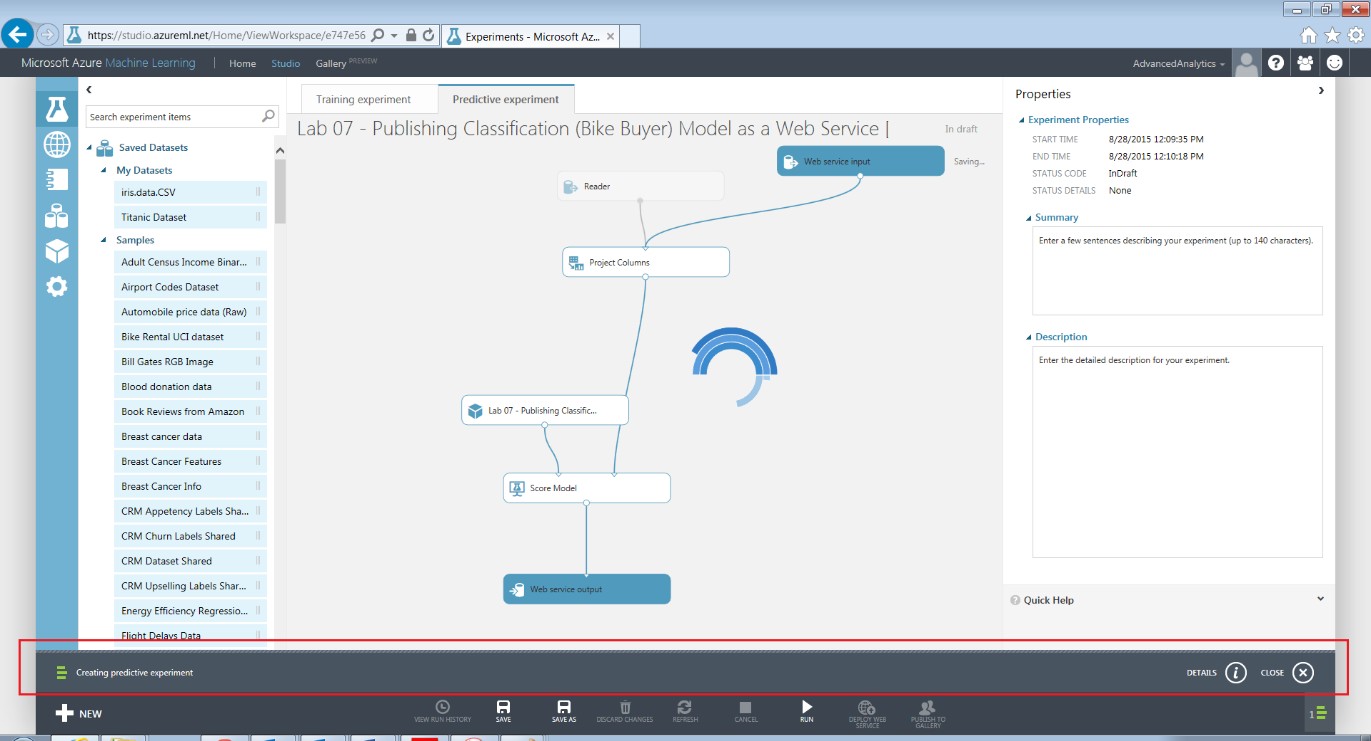
**Exercise 2: Setting up the Web Service**

1. In the bottom ribbon, click on the “SET UP WEB SERVICE” button and select “Predictive Web Service [Recommended]”. If the “Set up Web Service” button is greyed out for you, simply run the experiment again.

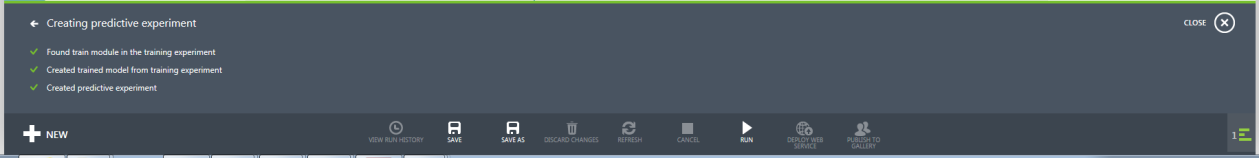


1. This makes the experiment model to optimize and splits the current experiment into two tabs, one is the “**Training Experiment**” and the other is the “**Predictive Experiment**”.
2. The work in progress screenshots are displayed below for your reference. Notice the

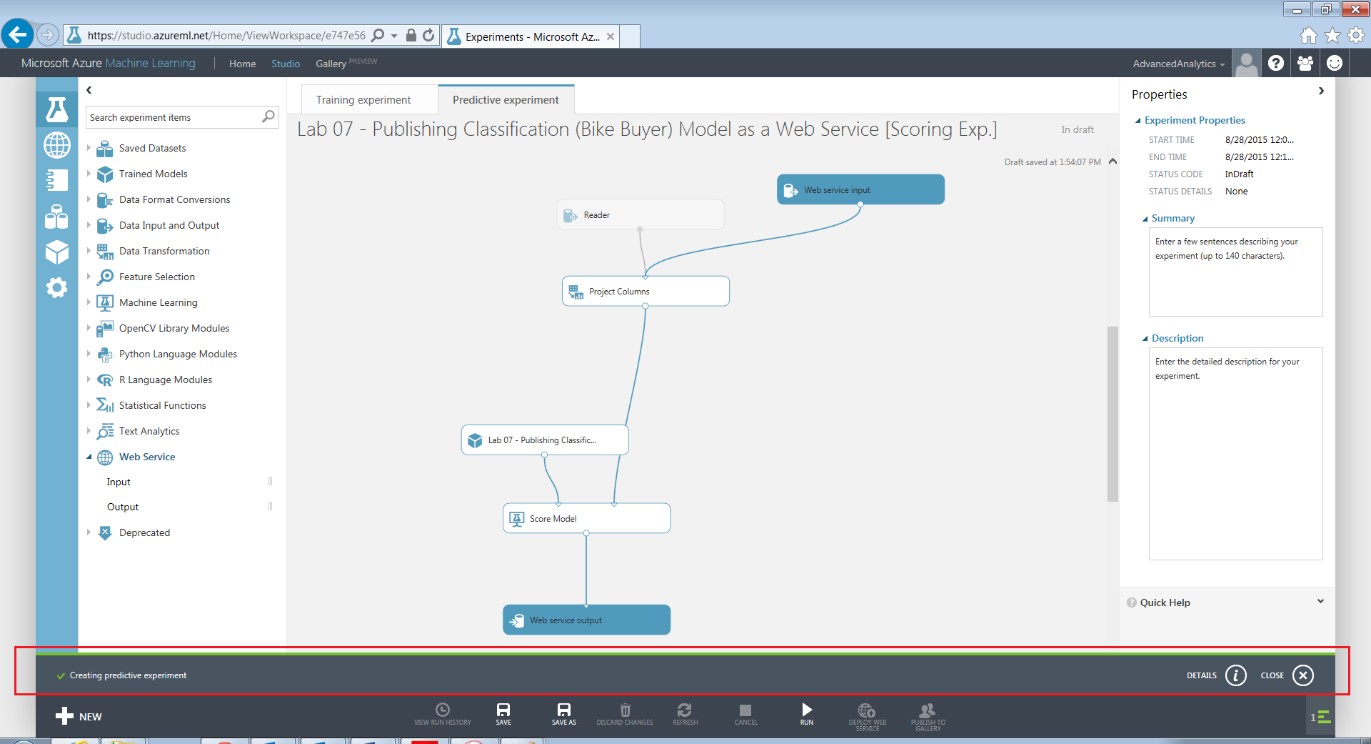
“Creating predictive experiment” status details as highlighted below at the bottom ribbon,



1. Click on the status details at the bottom ribbon and make sure it is completed successfully.



1. Once the “Creating predictive experiment” task is successfully finished, the experiment screen will look like below,



# Notice below changes that happened once the experiment is set up as a Web service.

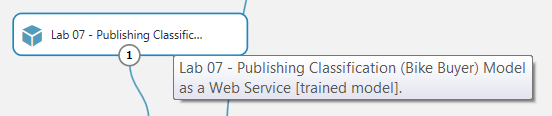
* 1. The current experiment is split into two tabs, one is the “**Training Experiment**” and the other is the “**Predictive Experiment**”.
  2. In the “**Predictive Experiment**” tab below changes occurred,
     1. The experiment name is suffixed with **[Predictive Exp.]**



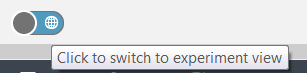
* + 1. 2 new components/modules got added, one is “Web service input” and the other is “Web service output”

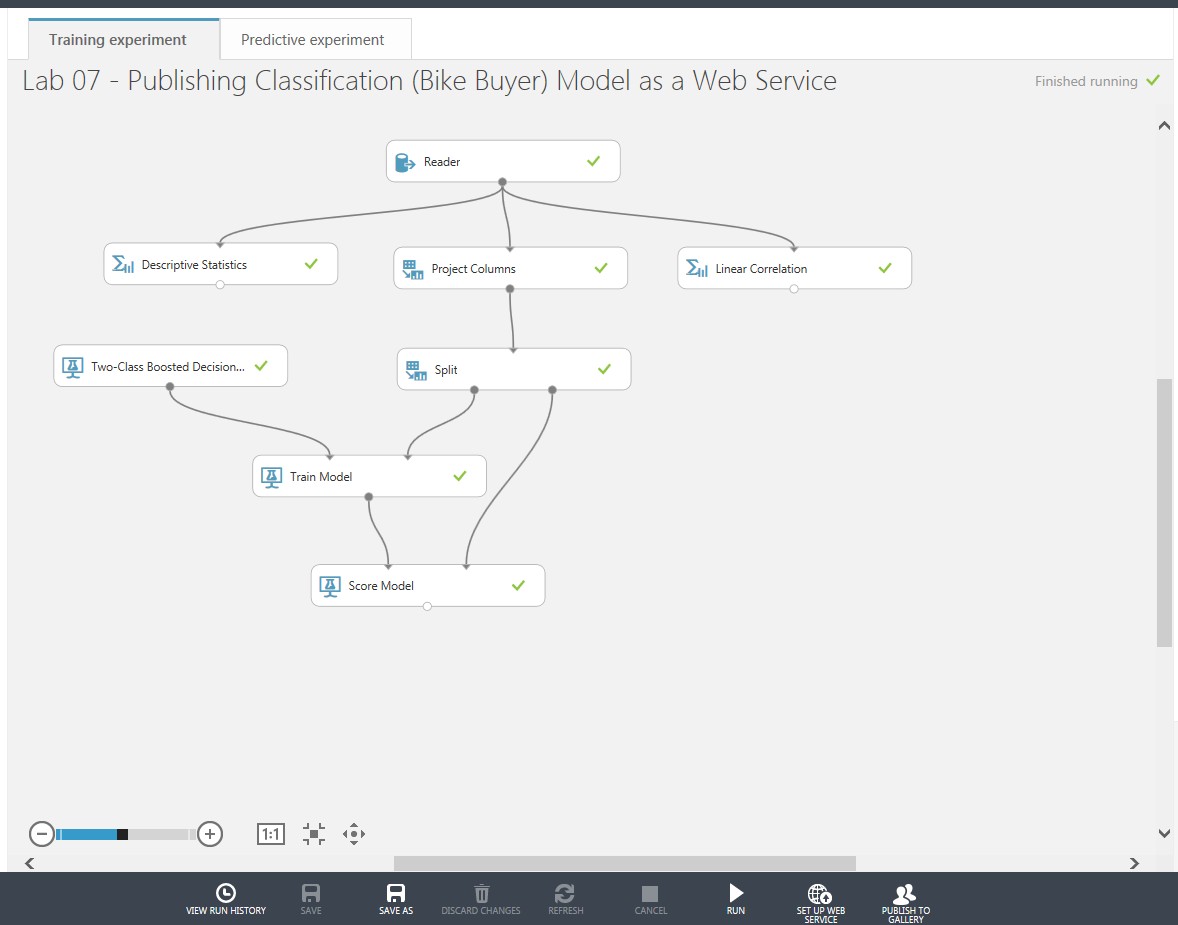
* + 1. Few of the modules got merged into one single component called as “trained model”



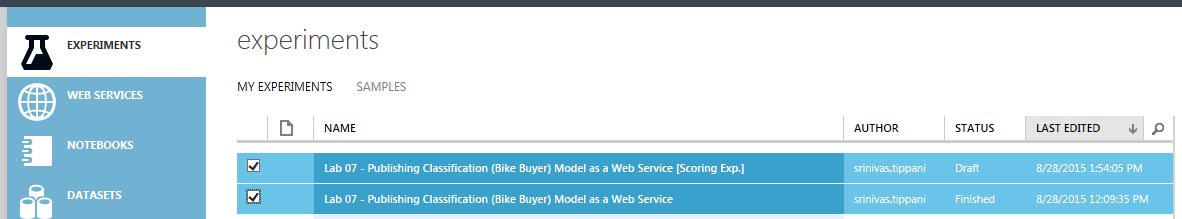
* + 1. A new switch got added to the experiment canvas pane, which can be used to switch between experiment view and web service view



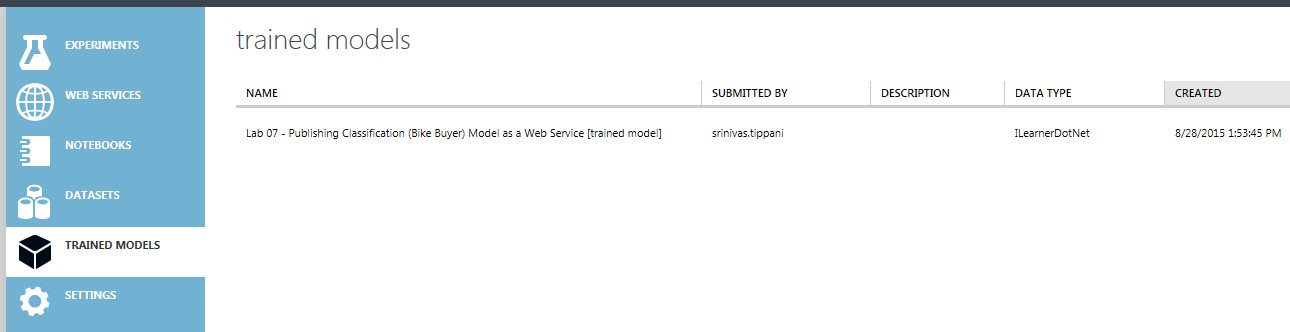
* 1. In the “**Training Experiment**” tab, the actual experiment is preserved “as-is”,



# Once the experiment is set up as a Web service, the actual experiment is SAVED AS a new experiment (as seen in the “Predictive Experiment” tab) with the experiment name suffixed with **[Predictive Exp],** as shown below,



* 1. Few of the modules that got merged into one single component (as seen in the “Predictive Experiment” tab) is SAVED AS a “Trained Model”, as shown below,

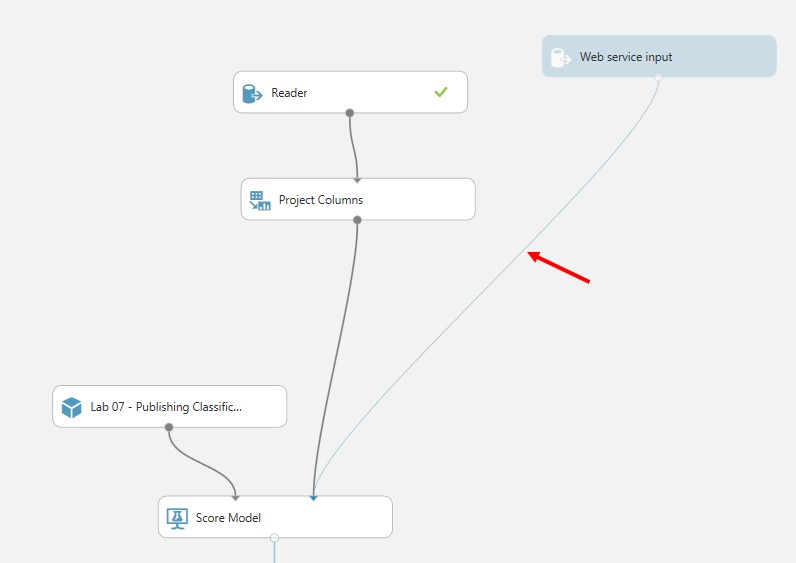


**Exercise 3: Deploying Web Service**

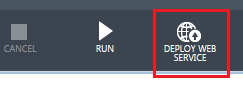
1. Open the Predictive experiment **“Lab 07 - Publishing Classification (Bike Buyer) Model as a Web Service [Predictive Exp.]”**
2. If you notice, the output port of “Web service input” is connected to input port of “Select Columns in Dataset” module.



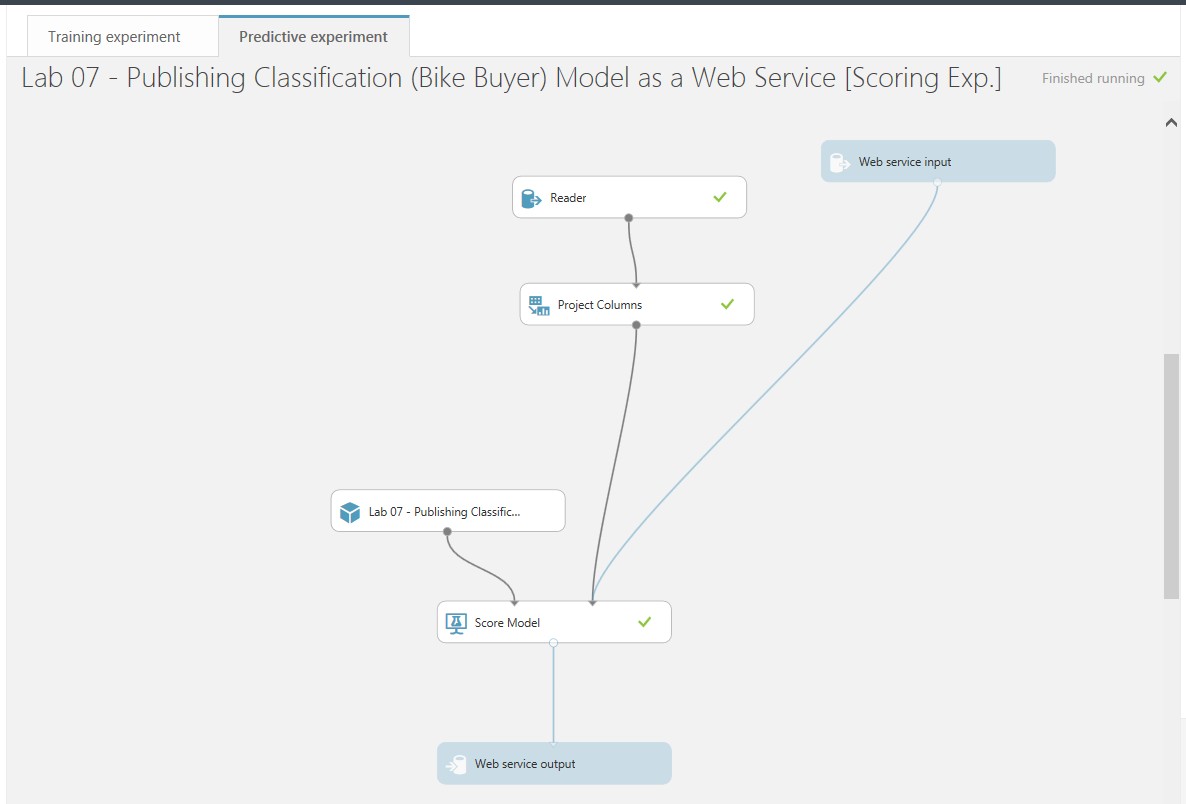
1. If you remember, in the “Select Columns in Dataset” we have excluded the unwanted columns (id, Yearly Take home Income) which are of less use. But, by having the output port of “Web service input” connected to input port of “Select Columns in Dataset” module, we still need to feed input values to these 2 unwanted columns, once the experiment is deployed as Web Service.
2. To remove these 2 unwanted columns from feeding input values, select and delete the output port of “Web service input” which is connected to input port of “Select Columns in Dataset” module by using the Delete key from key board, and connect the output port of “Web service input” module to right-side input port of “Score Model” module, as shown below,



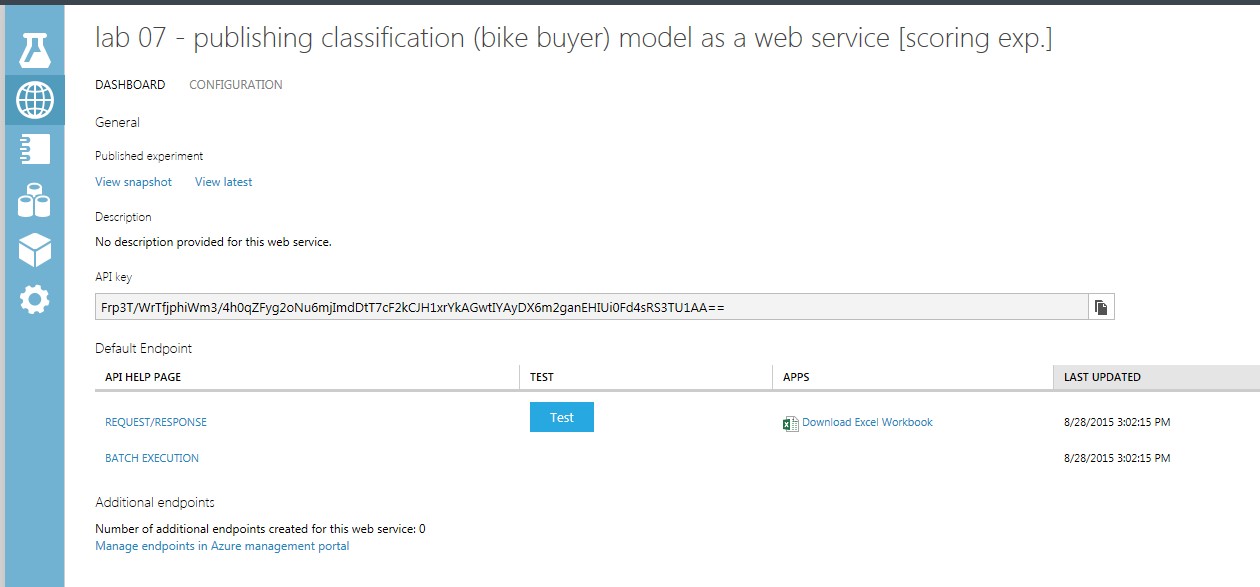
1. To deploy the experiment as a Web Service, first the experiment must be run. So, to run the experiment, in the bottom ribbon, click on the “RUN” button.



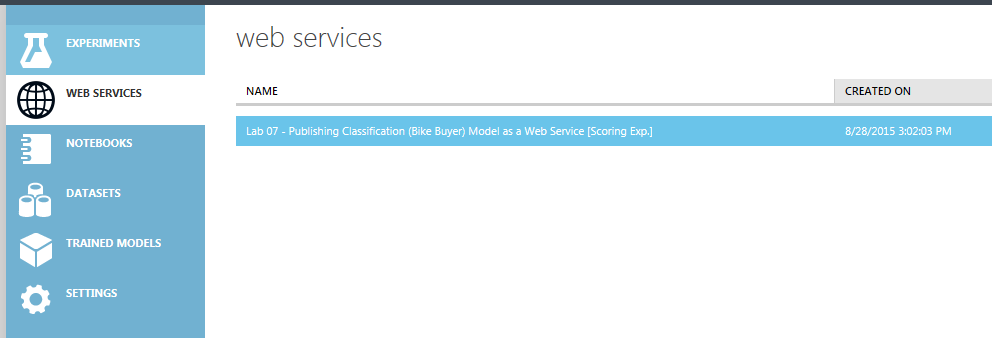
1. Once the experiment is successfully ran, it looks like below,

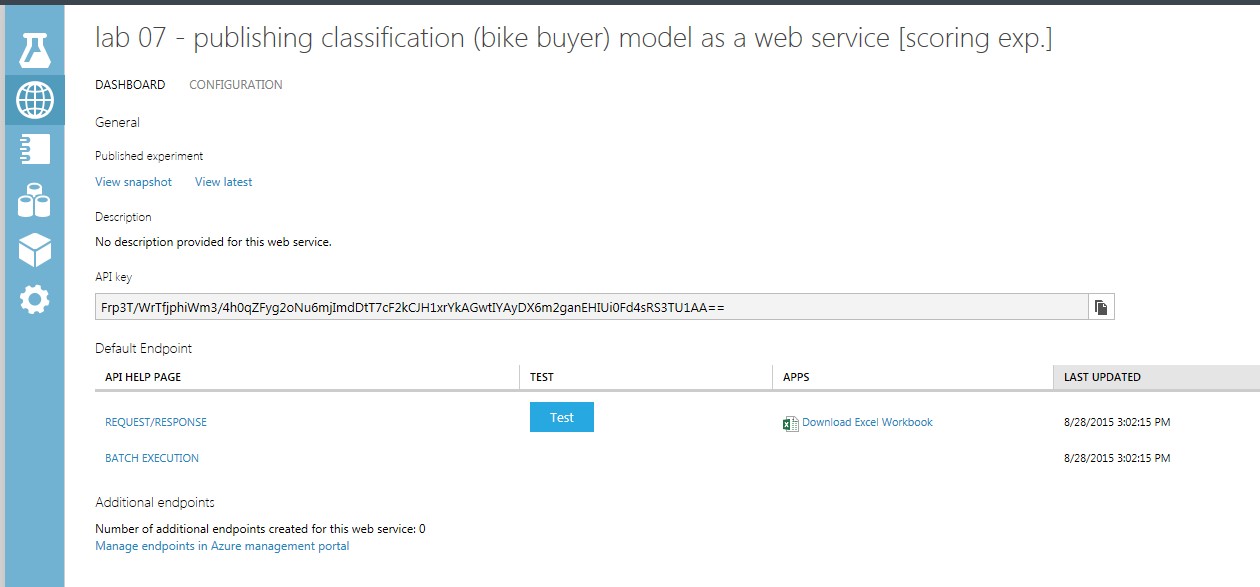


1. To deploy the experiment as a Web Service, in the bottom ribbon, click on the “**DEPLOY WEB SERVICE**” button.
2. Once the experiment is successfully deployed as a Web Service, the screen routes to

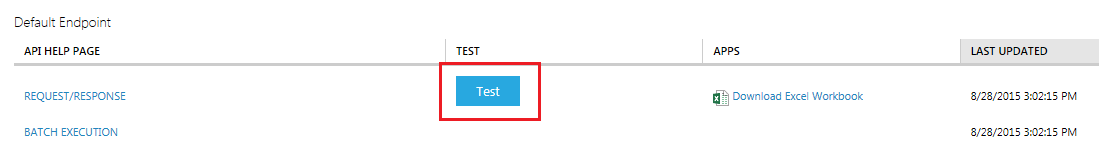


**Exercise 4: Testing your Model**

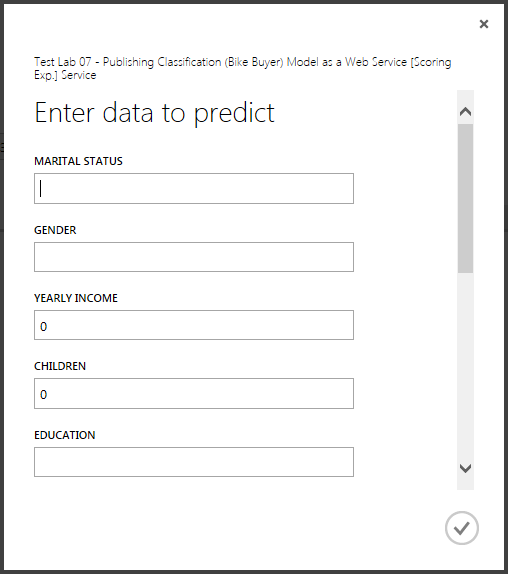
1. Open the Web Services pane.
2. Select the recently deployed experiment, “Lab 07 - Publishing Classification (Bike Buyer) Model as a Web Service [Predictive Exp.]”.



1. To Test your model, click on the **Test** button as shown below,



1. A form window pops up asking you to “Enter data to predict”.
2. The columns asking to be fed are the same columns that you inputted as schema to the “Web service input” module.



1. Input your own variables into the form. Remember that CASE MATTERS.

Ex: Marital Status as “Single” not “single” or “SINGLE”.

Refer to the Dataset provided to you “BikeBuyer.csv” for reference as to what values to be fed for each of the columns listed in the pop up window.

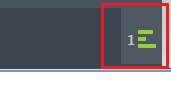
1. Let us take 2 below scenarios and feed the inputs accordingly,

**Scenario 1: -**

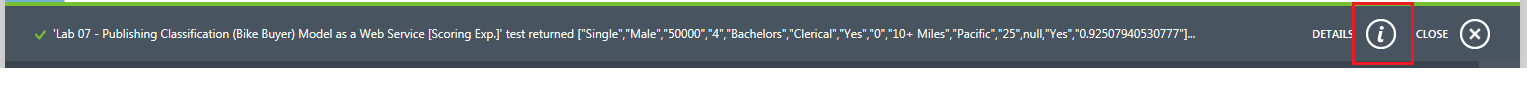
* 1. Feed below inputs,

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Marital Status | Gender | Yearly Income | Children | Education | Occu-pation | Home Owner | Cars | Commute Distance | Region | Age |
| Single | Male | 50000 | 4 | Bachelors | Clerical | Yes | 0 | 10+ Miles | Pacific | 25 |

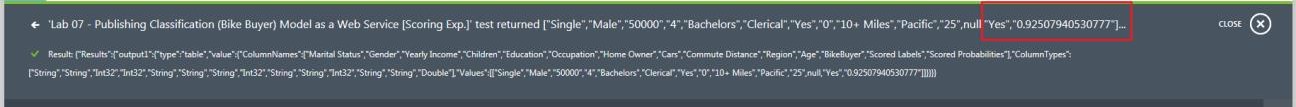
* 1. Submit the response. Look on the bottom right hand side for a loading box.



* 1. Click on it when it’s done loading. Click on details. Look at your result.



* 1. The above example shows a ‘Yes’ followed by a ‘0.9250’ meaning it predicted, from the parameters given, that this person would have a 92.5% chance of buying a bike.

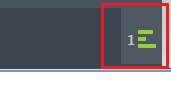


**Scenario 2: -**

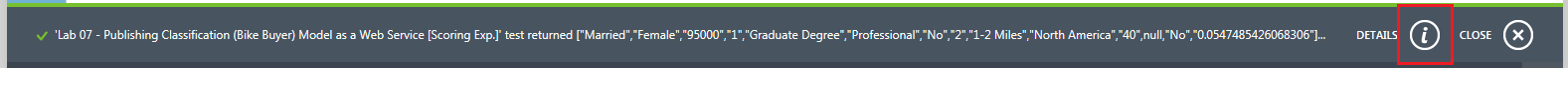
1. Feed below inputs,

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Marital Status | Gender | Yearly Income | Children | Education | Occu- pation | Home Owner | Cars | Commute Distance | Region | Age |
| Married | Female | 95000 | 1 | Graduate Degree | Pro-fessional | No | 2 | 1-2 Miles | North Ame-rica | 40 |

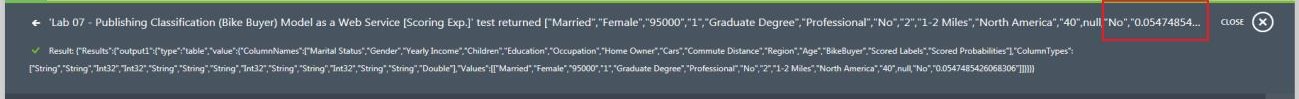
1. Submit the response. Look on the bottom right hand side for a loading box.



1. Click on it when it’s done loading. Click on details. Look at your result.



1. The above example shows a ‘No’ followed by a ‘0.0547’ meaning it predicted, from the parameters given, that this person would only have a 5.47% chance of buying a bike.



1. Play around with the parameters to see how your model performs for various input combinations.