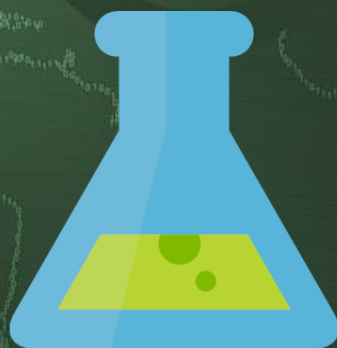


# Azure Machine Learning Studio: An Unleashed Guide

**Author: Leila Etaati**  
**Edition: One, February 2019**



PUBLISHED BY

RADACAD Systems Limited

<http://radacad.com>

24 Riverhaven Drive, Whangaparoa, Auckland, New Zealand

Copyright © 2019 by RADACAD All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

**Cover by: pixabay.com**

## About the book; Quick Intro from Author

Azure Machine learning has been introduced in 2014. By seeing a demo in SQL PASS Summit, I get interested in this product. From that time, I start to work with and demonstrating in different conferences. After a while, I start to write some weblog post about it. In this book, I gathered all these posts in one book. Totally I wrote about 10 blogs on it in different time from 2017. The book contains information about Azure ML Studio environment, how to use algorithms and create predictive analytics, how to transform data and so forth. You can start reading this book with no prerequisite. It is better to follow the provided sequences.

## About Author

[Leila](#) is the First [Microsoft AI MVP](#) in New Zealand and Australia, She is a Data Platform Microsoft MVP as well. She has PhD in Information System from the University Of Auckland. She is the Co-director and data scientist in [RADACAD](#) Company with more than 100 clients in around the world. She is the co-organizer of [Microsoft Business Intelligence and Power BI Use group \(meetup\)](#) in Auckland with more than 1200 members, She is the co-organizer of three main conferences in Auckland: [SQL Saturday Auckland](#) (2015 till now), [Difinity](#) (2017 till now) and [Global AI Bootcamp 2018](#). She is a Data Scientist, BI Consultant, Trainer, and Speaker. She is a well-known International Speakers to many conferences such as Microsoft ignite, SQL pass, Data Platform Summit, SQL Saturday, Power BI world Tour and so forth in Europe, USA, Asia, Australia, and New Zealand. She has over ten years' experience working with databases and software systems. She was involved in many large-scale projects for big-sized companies. She also AI and [Data Platform Microsoft MVP](#). Leila is an active Technical [Microsoft AI blogger](#) for RADACAD.



## Who Is This Book For?

This book is designed for BI Developers, Consultants, Data scientists who wants to know how to develop machine learning solutions with Azure Machine Learning Studio. BI Architects and Decision Makers who wants to make their decision about using or not using Machine Learning for their BI applications. Business Analysts who want to get better insight on data and learn tricks of how to apply machine learning on specific data. The book titled "Azure Machine Learning Studio", and that means it will cover wide range of readers. I'll start by writing 100 level and we will go deep into 400 level at some stage. So, if you don't know what Azure ML Studio is, or If you are familiar with machine learning but want to learn how to use Azure ML Studio, this book able to show you the main process.

## Upcoming Training Courses

Leila runs different Microsoft Machine Learning training courses both online and in person. RADACAD also runs Power BI and SQL Server courses ran by Reza Rad. Our courses run both online and in person in major cities and countries around the world. Check the schedule of upcoming courses here:

<http://radacad.com/events>

<http://radacad.com/advanced-analytics-training>

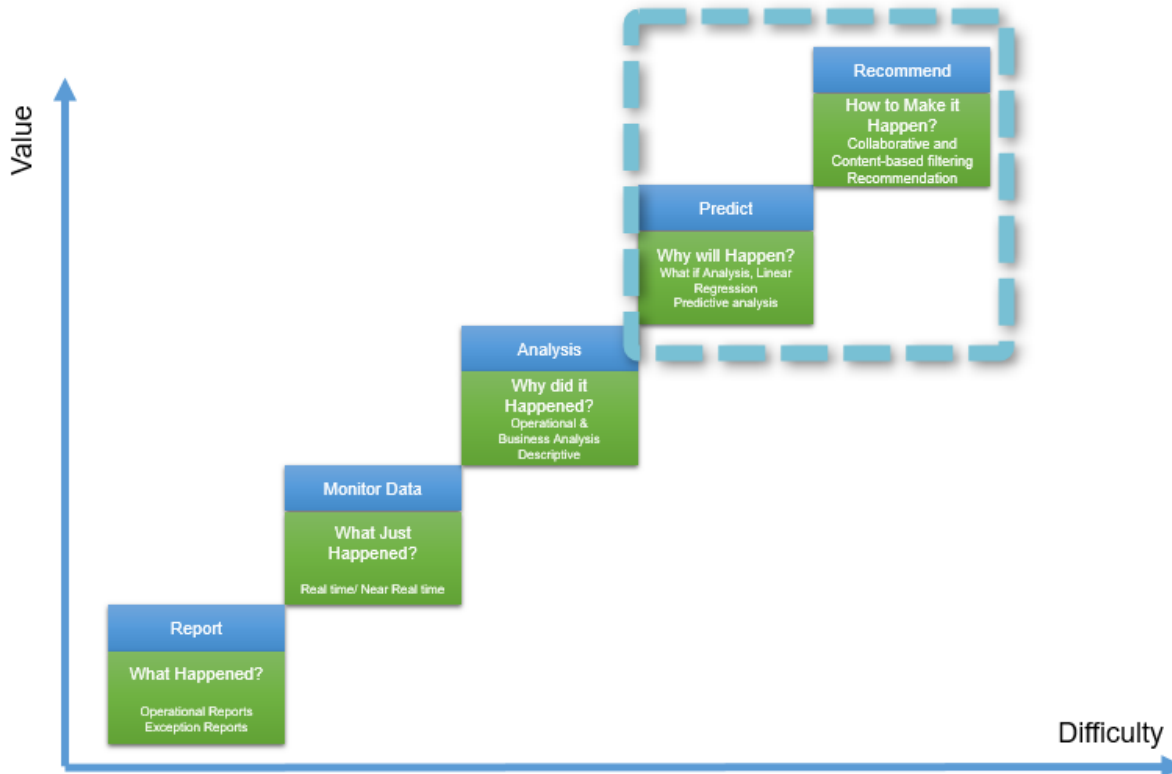
# Table of Content

## Contents

Chapter 1: Introduction.....	8
Chapter 2: Azure ML Studio Environment.....	15
Chapter 3: Data Transformation in R .....	28
Chapter 4: A Machine Learning Prediction Scenario – Data Cleaning .....	40
Chapter 5: A Machine Learning Prediction Scenario – Feature Selection.....	49
Chapter 6: A Machine Learning Prediction Scenario (1).....	61
Chapter 7: A Machine Learning Prediction Scenario (2).....	70
Chapter 8: Tune Parameters: Machine Learning Prediction .....	76
Chapter 9: Cross Validation: Machine Learning Prediction.....	81
Chapter 10: Create Web service from Models.....	92

# Chapter 1: Introduction

Published Date: March 27, 2017



In this book, I will talk about Microsoft cloud machine learning: Azure ML Studio. I will explain the main components and concepts of Azure ML Studio. In the first chapter, I will talk about Machine Learning concepts and Azure ML.

## What is Machine Learning:

Machine learning according to Wikipedia is:

*"subfield of computer science that gives computers the ability to learn without being explicitly programmed"*



The main concept comes from learning from data and then for a new series of data, predict based on the past data behavior.

The best example is: handwriting recognition in a Post Office.



The computer will be fed by many different handwriting styles. For instance, for the word "Referred" we may have different ways of writing it (see below image). A machine learning program will learn from all different writing styles, then in the new set of data, it should be able to distinguish it. So, the computer program will **learn** that the word "referred" can be written in different ways. Hence, for future letters, the computer program will be able to distinguish different varieties of writing "referred".

referred  
referred  
referred

referred  
referred  
referred

referred  
referred  
referred

referred  
referred  
referred

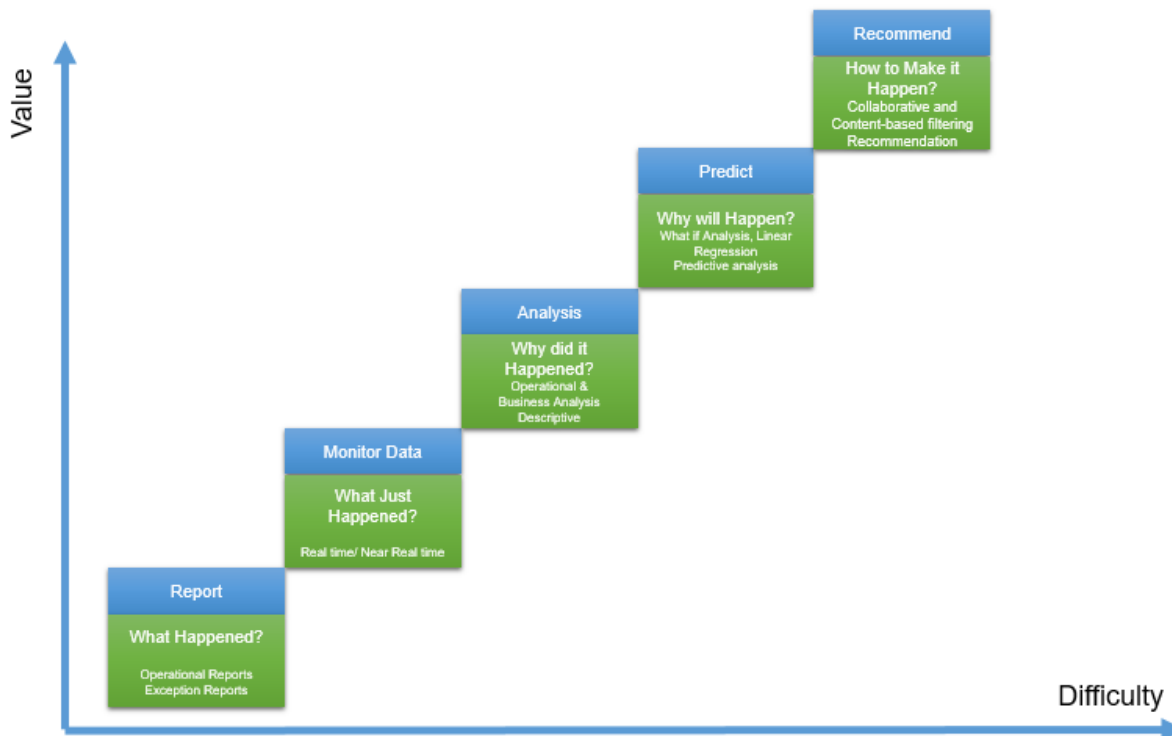
referred  
referred  
referred

referred  
referred  
referred

referred  
referred  
referred

referred  
referred  
referred

Moreover, machine learning is a new trend in BI. Before this era, we mainly focused on “What is happened”, “Why it happened”, or “What is happening now”. The new trend that we are looking at is: “What **will** happen” and “**How to make it happen**”



The first level (What happened) has been used for many years in BI systems. Which is about checking what happened in the past without any level of analytics, while the second level fetches the same data, but it will look at the real-time or near real-time data. It can be an example of the internet of things. In the third level, the main aim is to see “why it happened” so we focus is on what if analysis.

The fourth level mainly is about the “What will happen” which machine learning will be used here. It is not an easy task but it worth the effort especially for future decision making. Finally, the last part is about the Recommendation, that based on the specification of customers what action/products/ so forth good for them.

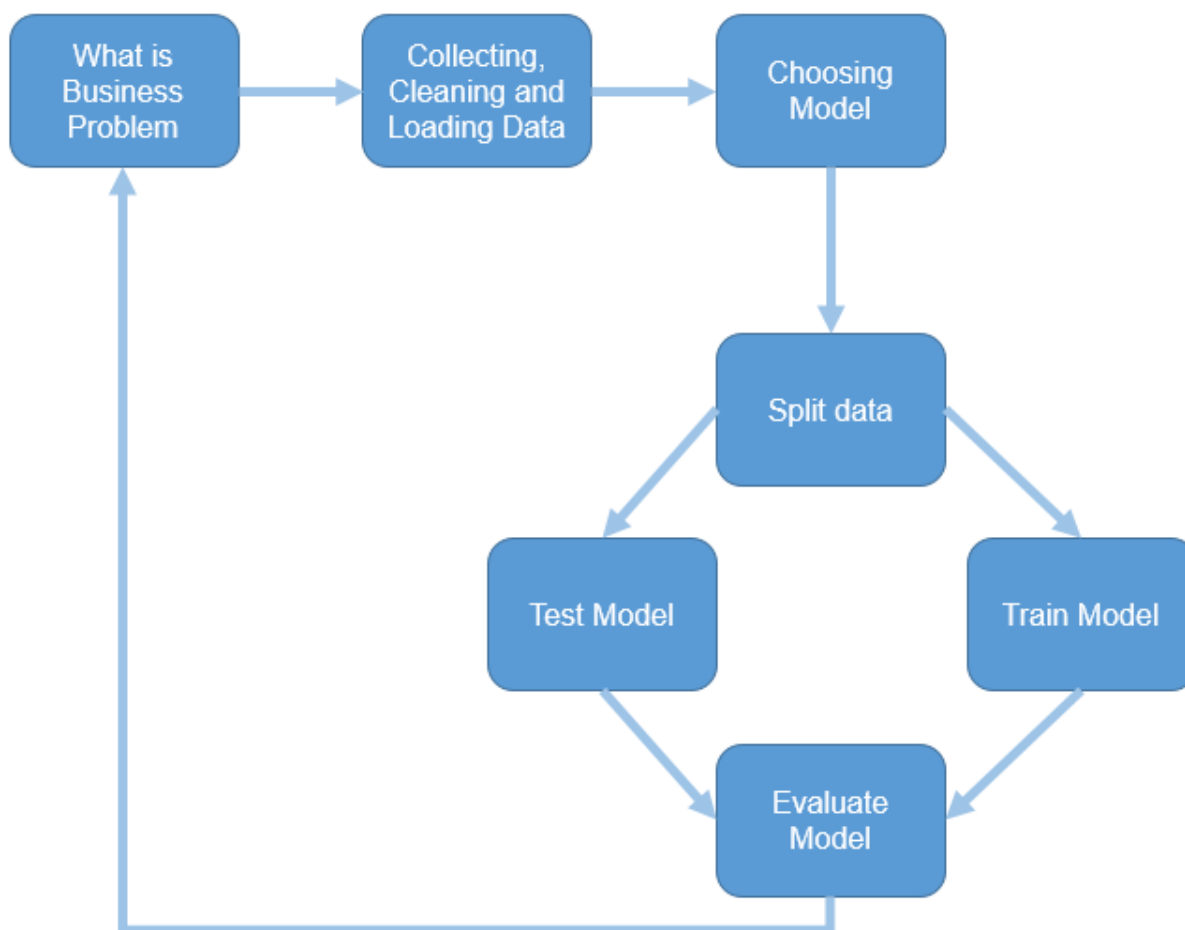
For the first and the third level we can use BI tools such as Microsoft Power BI. The good example for the second level is “IoT” or using “Gateways” in Power BI.

However, the main aim of the current series is more on the fourth and the fifth level of analysis which is predictive and prescriptive analytics.

## Machine Learning Cycle

To do machine learning, we should follow a specific cycle as below:

The first step is to identify the business problem such as: sale prediction, customer churn and so forth. The next step is about identifying the relevant data, what attribute has more impact on the problem, gather data, clean data and load data. In the second step, the ETL process may be applied. Finally, after gathering data and data wrangling, we have to choose the relevant algorithm based on the problem nature (will be explained later in this series). Finally, we split data some part for the training model and some other for testing the mode. The model will learn from past data to better predict the upcoming data. Then, we evaluate the created model and deploy it.



In some cases, the model evaluation is not satisfactory, hence we must reconsider the business problem, collect other data or choose another model.

This process applies to any other machine learning platform.

## Azure Machine Learning

Azure ML Studio is a cloud machine learning platform. It is part of the Cortana analytics suite. Azure ML Studio make the process of creating predictive, descriptive and prescriptive analysis much easier.



Azure ML Studio is in “Machine Learning and Analytics” part of Cortana.

The main specifications of using Azure ML Studio are as below;

1. Web-based, accessible from anywhere
2. If you are using Cortana, for example Azure SQL database, Azure Virtual Machine and so on, it so easy to create analytics on it. For instance, you can get data from SQL Data Base in Azure and apply some analytics on them, then store the result in Azure again. Moreover, if you use Event hub or IoT hub, you can easily create analytics on live data (see

<http://radacad.com/part-4-live-streaming-weather-station-with-cortana-analytics-online-data-analytics-with-azure-ml>)

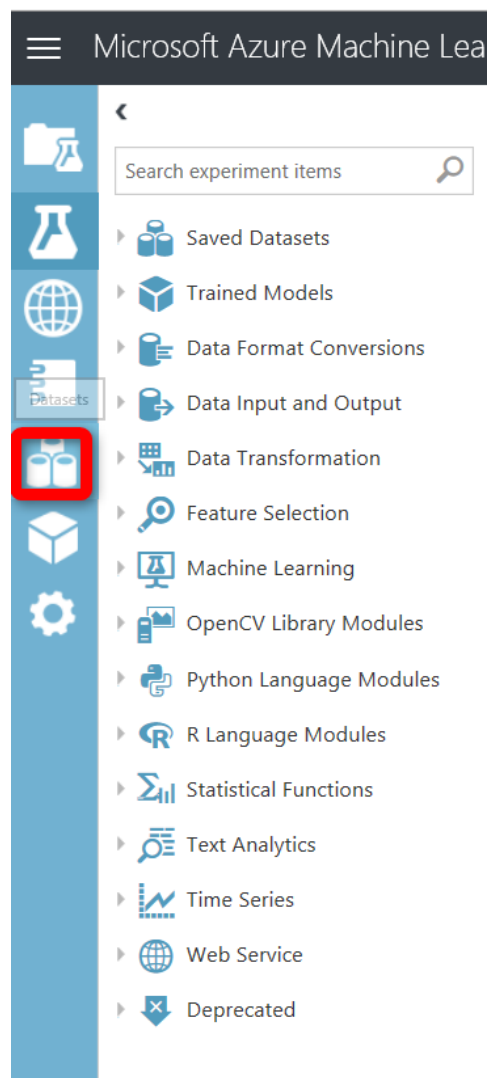
3. Create a fast prototype for Machine Learning. Azure ML Studio has a group of algorithms which easy to use (will explain in next parts), the drag and drops environment will make it easy to create a fast prototype.
4. Flexibility in using the algorithm,25 algorithms that already have been used by the Xbox team and Bing search team. also there is a possibility to write your R or Python Codes there
5. Create web service out of the created model by a few clicks.

And so many others.

In the next chapter I will show the Azure ML Studio environment and its main components.

# Chapter 2: Azure ML Studio Environment

Published Date: March 28, 2017



In the previous chapter, the main concepts of Machine Learning have been explained very briefly. However, if we want to talk about Machine Learning, it needs to read a whole book. In the second part, I am going to show the main Azure ML environment and its essential components.

Azure ML has two types of subscriptions: Free, and Standard. In this chapter, I am using free subscription. The extensive explanation of the Standard subscription will be explained in the last

part of this series. To sign into the Azure ML portal, you can use any of your email accounts: Gmail account, Microsoft account, or company email. using this URL to log in to azure ML Studio:

<https://studio.azureml.net/>

First, you should log in to the signing page as below

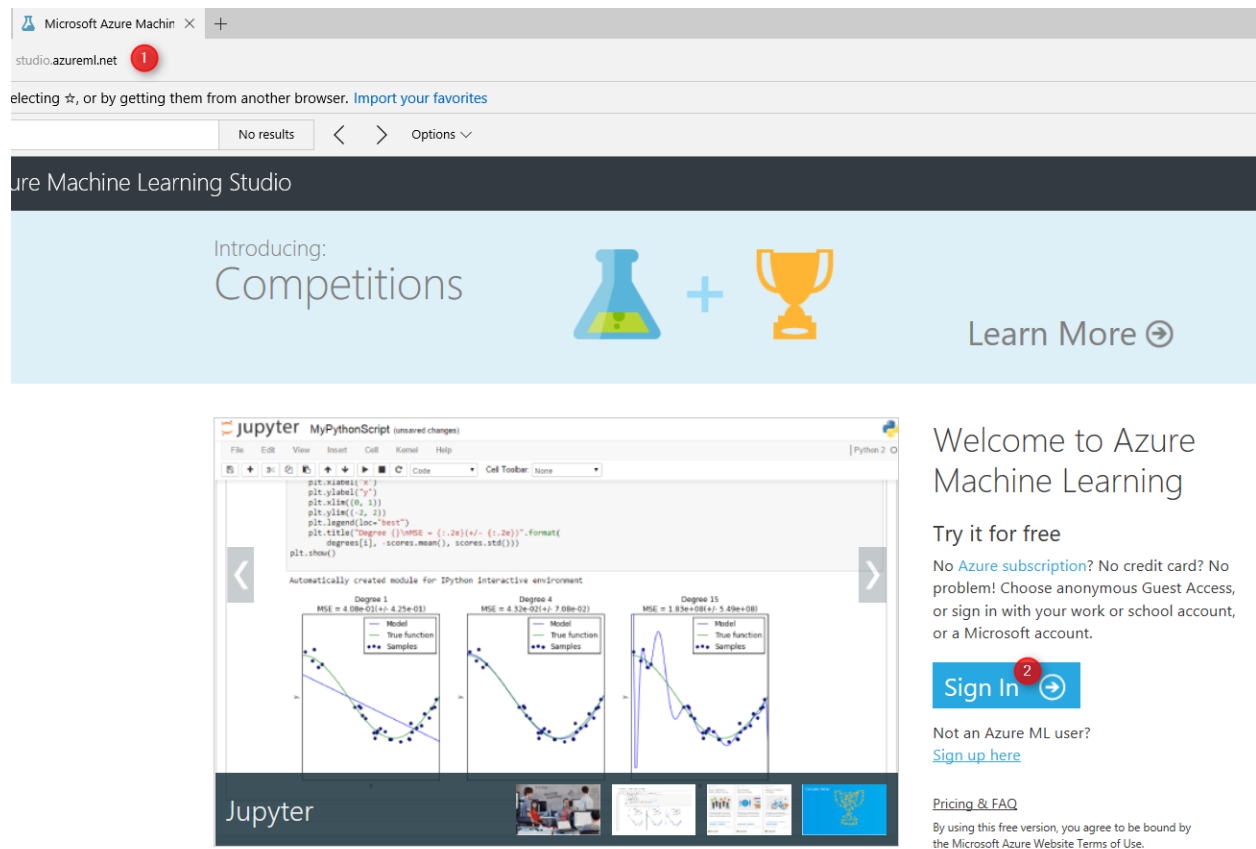


image 1. Login Page

Then, after sign into the Azure portal, you will see below page. Azure ML studio has different components. The first component is "Experiments". The experiment is the first place that we will use for creating a machine learning project (number 1). to Create a new experiment, we have to click on the "New" icon in the bottom of the page. The list of created experiments will be shown in the middle of the page (Number 3). If you want to delete an experiment, you first should select it,



then click on the "Delete" icon (number 4). Moreover, for the copy-specific experiment, select it and then click on the "Copy experiment" icon (number 5).

In the left side of the window, we have other components such as "Web Services", "Notebooks", "Datasets", "Trained Models" and "Settings".

"Web services" is a place that the created web services will be shown there and you able to see them (I will explain the process of creating web services in the next series). Also we have the "Notebook" component, for Python users (number 7). The imported dataset and new datasets will be accessible via the "Datasets" icon (number 8). Moreover, the "Trained Models" also store the created models that we can use in other applications (will explain it in the next parts). And finally, "setting component" that help us to arrange the environment.

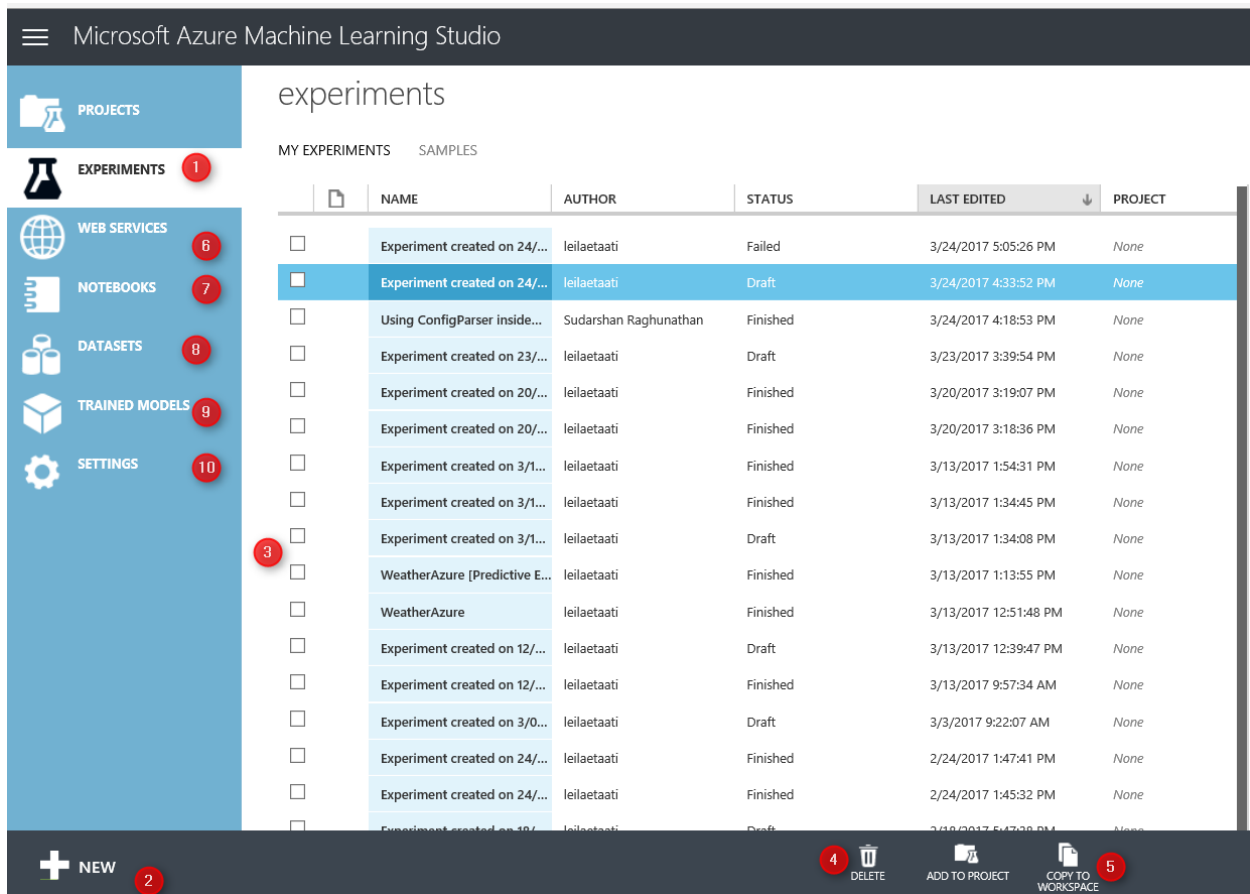


Image 2. Main Page

In this series I will talk about “Experiments”, “Datasets”, “Web Services”, “Trained Models”, and the “Setting”.

In this chapter, we first look at the “experiment”. In the experiment, we able to create a new Machine Learning Module. In the experiment area we have a drag and drop environment, that we can put items from the menu on the left side to the middle of the screen (Number 1). Moreover, we able to put some more description of the experiment we are creating on the right side of the screen (number 2). Each Experiment could have a specific name on top of the page (Number 3) that has been assigned by the user. All the main components and tools for creating a machine learning module have been put on the left side of the screen (Number 4). We are also able to “Save/ Save as” the component (number 5). Finally, after creating a model, we can “Run” the experiment.

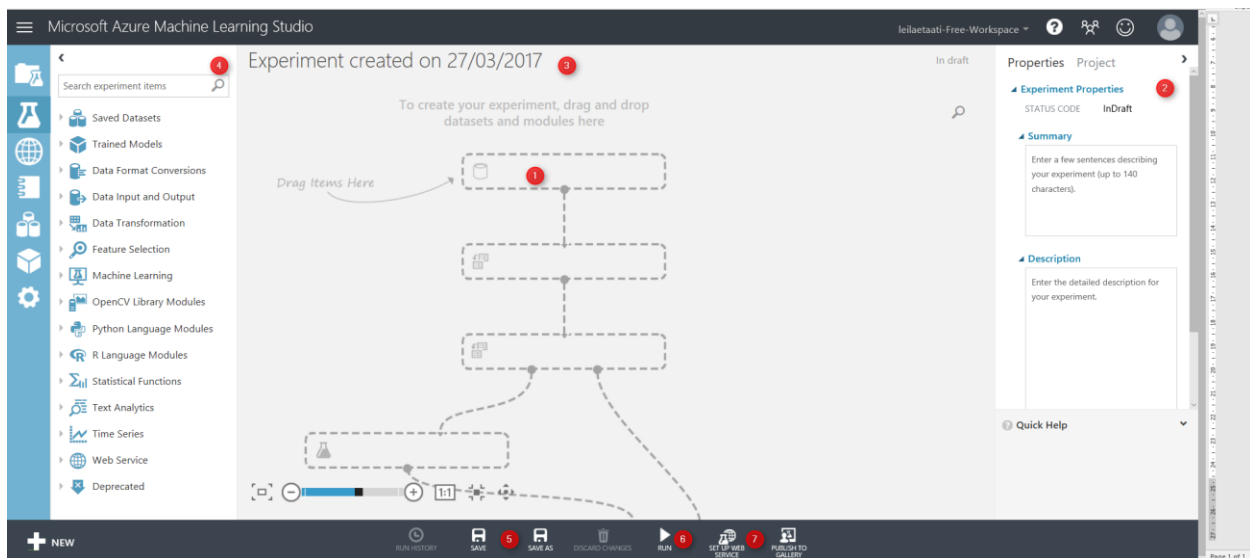


Image 3. Experiment

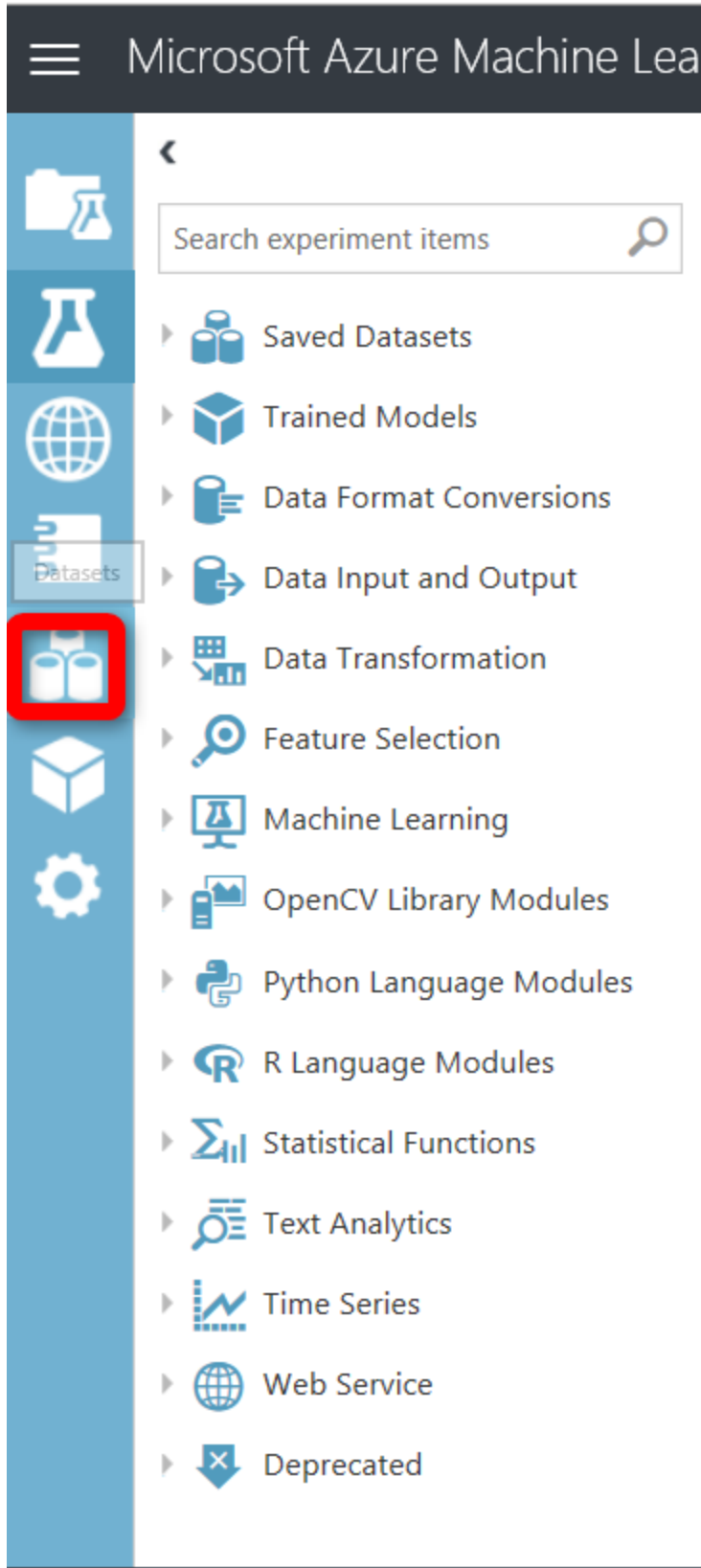
According to Chapter1 of this series, the first step after identifying the business problem, is about data gathering, data cleaning, data filed selection and splitting data.

To start a machine learning process, we have to get data first from resources.

There are a couple of options to fetch data.

### 1. Import Data set

the first approach to import data, first click on the "database" icon on the left side of the screen

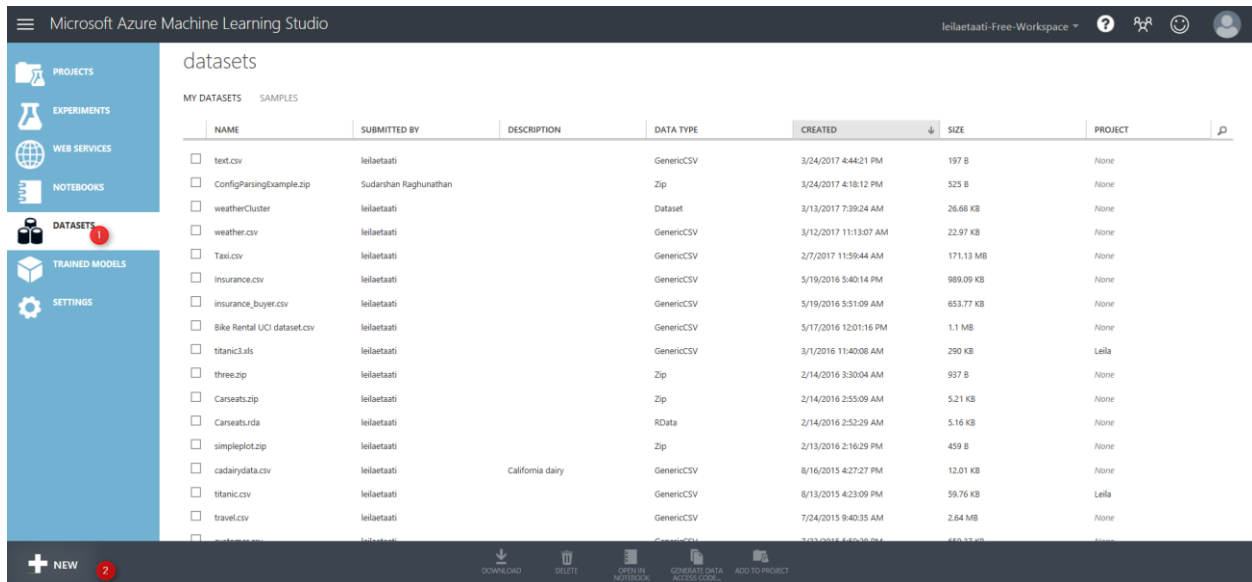


Microsoft Azure Machine Learning Studio

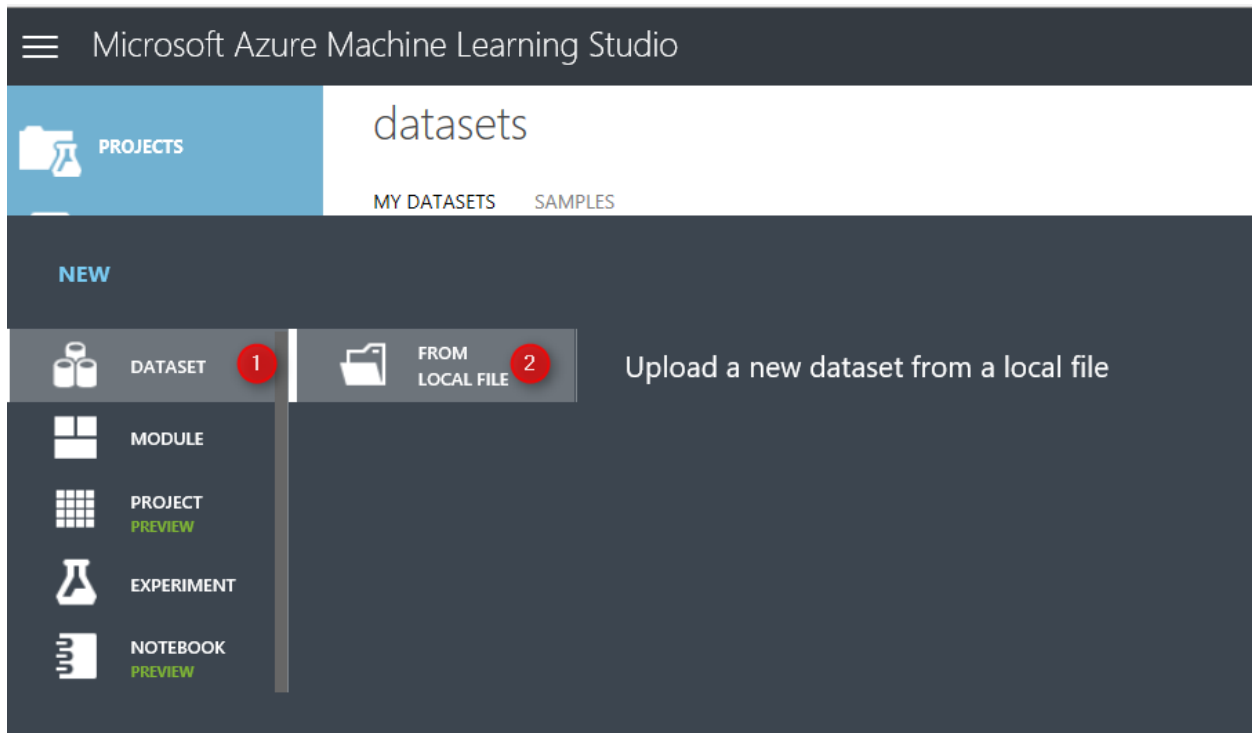
Search experiment items

- ▶ Saved Datasets
- ▶ Trained Models
- ▶ Data Format Conversions
- ▶ Data Input and Output
- ▶ Data Transformation
- ▶ Feature Selection
- ▶ Machine Learning
- ▶ OpenCV Library Modules
- ▶ Python Language Modules
- ▶ R Language Modules
- ▶ Statistical Functions
- ▶ Text Analytics
- ▶ Time Series
- ▶ Web Service
- ▶ Deprecated

then in the Data set window, first click on the data set icon in the left side (Number 1), then on the “New” icon on the bottom of the page.



Then, below windows will be shown up. To import the dataset from your local pc, you should click on the “dataset” icon (Number 1), then click on the “From Local File” icon (Number 2)



Then, you able to upload a new data set into the Azure ML experiment. Just browse your pc to select the data set. There are some limitations on data set type as just able to import directly from local pc just: CSV, TSV, TXT, ARFF, Zip and RData data types.

After selecting datatype (number 2), approve it (number 3). You will able to see the data set in the Azure ML environment.

---

x

## Upload a new dataset

**SELECT THE DATA TO UPLOAD:**

Browse... 1

This is the new version of an existing dataset

**ENTER A NAME FOR THE NEW DATASET:**

Select a dataset type...

Generic CSV File with a header (.csv)

Generic CSV File With no header (.nh.csv)

Generic TSV File with a header (.tsv)

Generic TSV File With no header (.nh.tsv)

Plain Text (.txt)

SvmLight File (.svmlight)

Attribute Relation File Format (.arff)

Zip File (.zip)

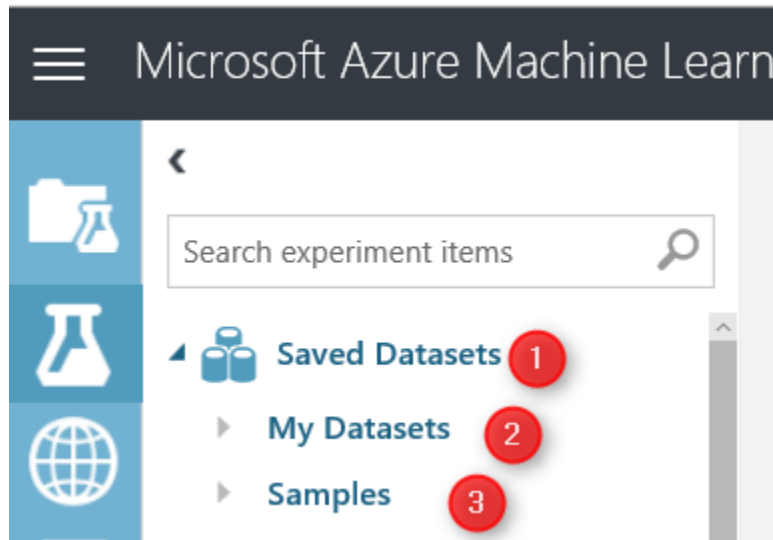
R Object or Workspace (.RData)

2

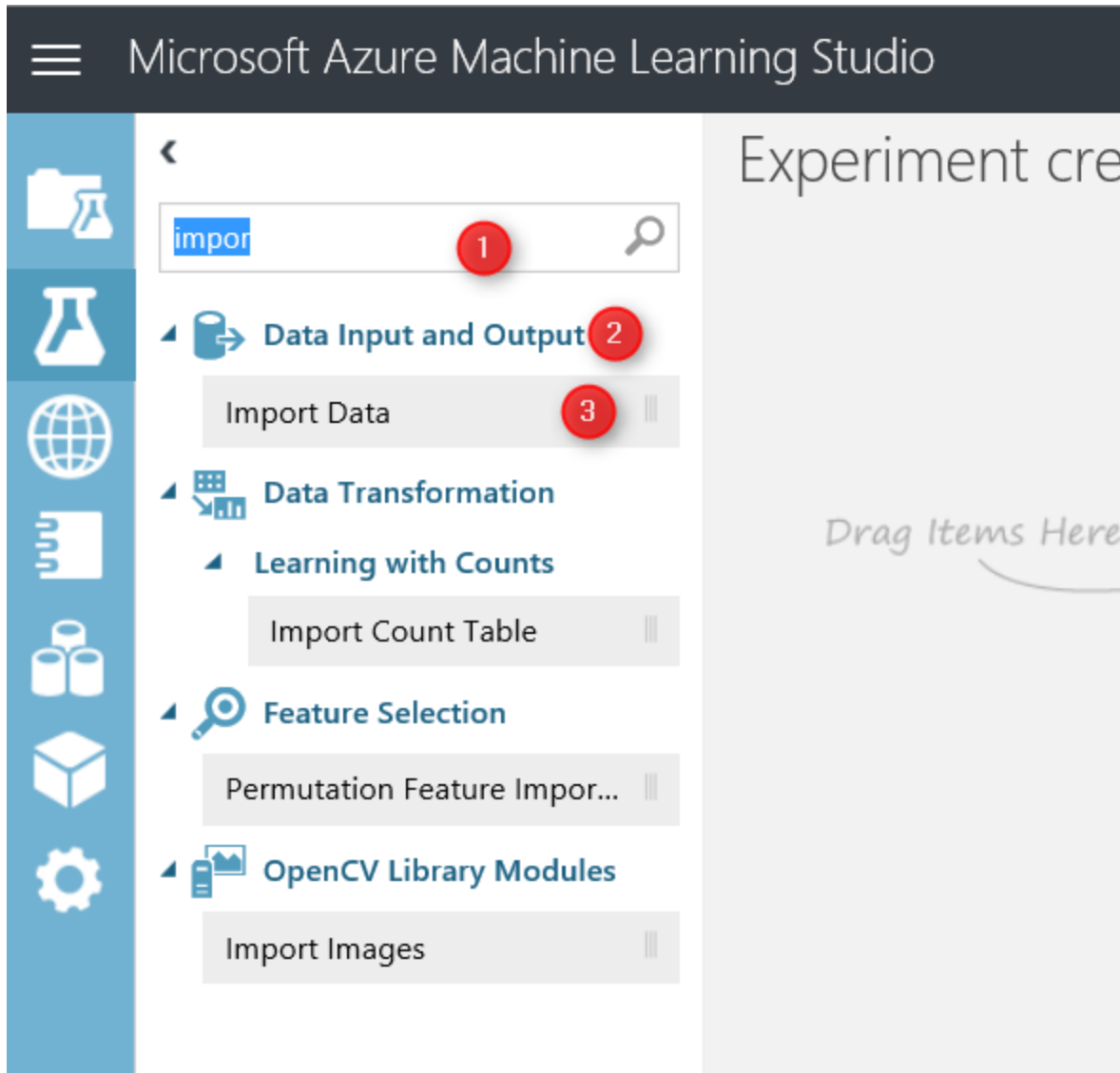
3  
✓

---

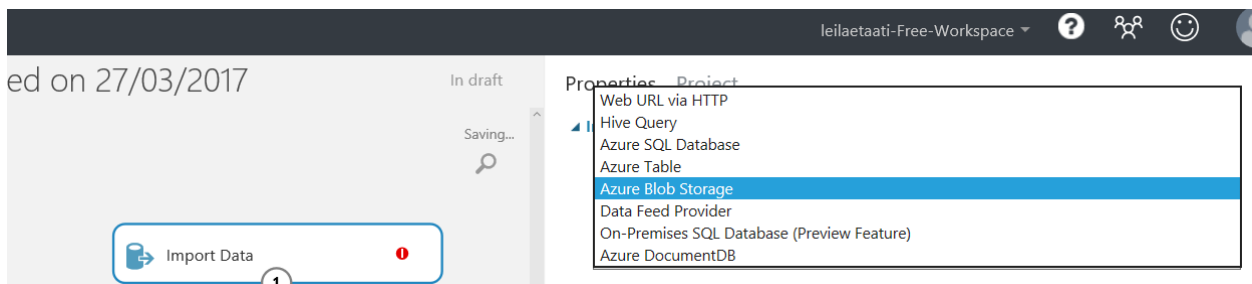
In the Azure ML experiment area, you will see a "Saved Datasets" icon, also under the "My Datasets" you will find uploaded data set. However, there are some predefined data set in the system as Microsoft provided them for users (Number 3).



Importing data from local pc is not the only way for importing data, the other way is using component "Import data". just type word "import" in the search bar in number 1. You will see under the "data Input and Output" component, we have a module name "Import data" (Number 2 & 3). Just Drag it and drop it in the middle of the screen in Experiment area (Number 4)



After putting Module " Import data" in the experiment, we able to identify the main source of data from a variety resources in the cloud and web resources (see below image)



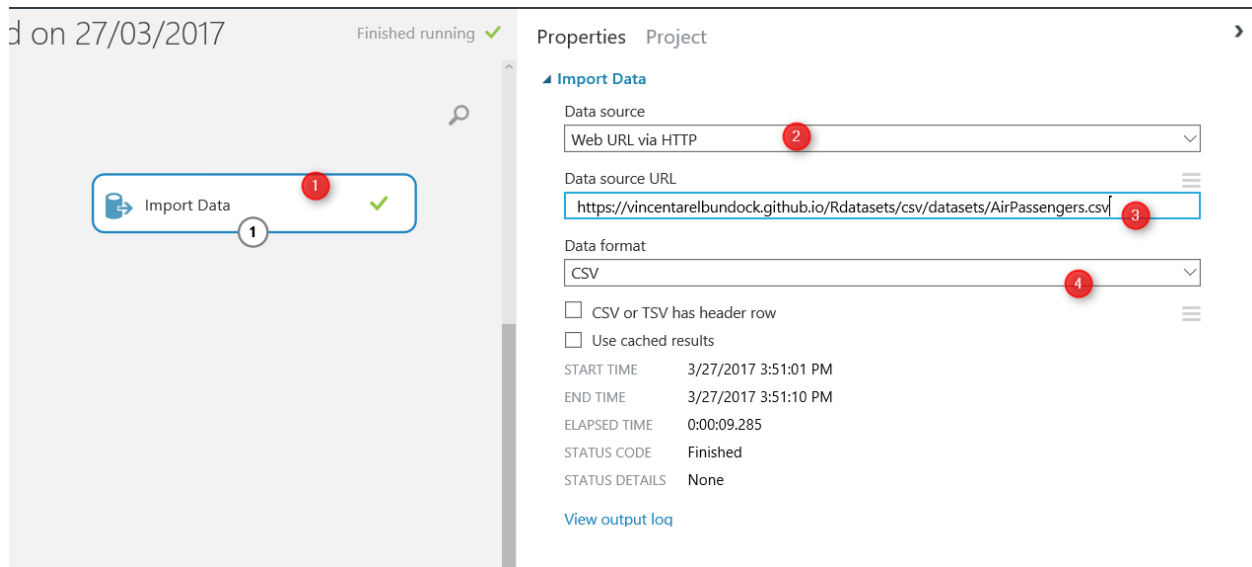


By using "Import data" you are able to get data from different resources as has been shown in the above image. Many of them are part of the Azure Cortana suite (Azure Blob, Azure SQL DB, Azure doc, and Azure Table) however there is a possibility to import datasets via web URL (first option) or from a web service (data feed provider).

For example, I want to import data from a web URL:

<https://vincentarelbundock.github.io/Rdatasets/csv/datasets/AirPassengers.csv>. This is a CSV file.

First, I click on the Import data module, then in the "data source" I will choose "Web URL via HTTP" (number 2), following, in the front of the "Data Source URL", put the web link URL. Finally, in "data format" identify the data format as CSV (or another format). Then just run the experiment.



The screenshot displays the Azure Machine Learning Studio interface. On the left, a workflow canvas shows an "Import Data" module with a green checkmark and a red circle labeled "1" at its bottom. The top of the canvas indicates "Finished running" with a green checkmark. On the right, the "Properties" pane for the "Import Data" module is visible. It shows the following configuration:

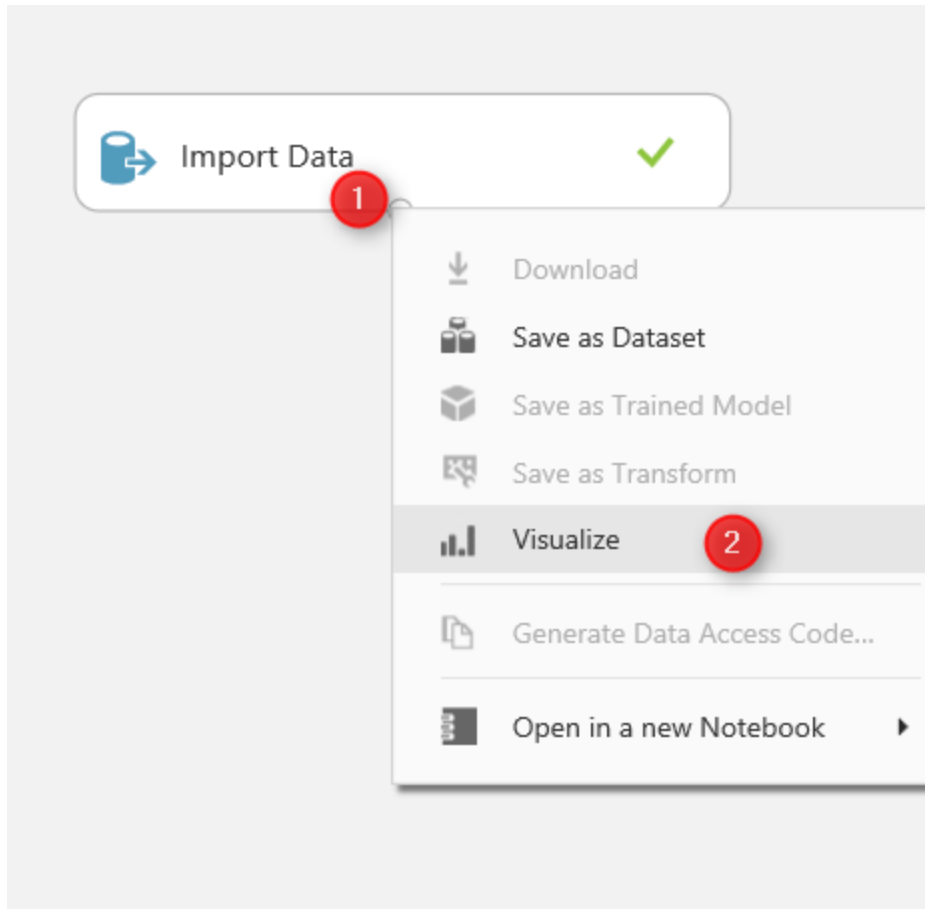
- Data source:** Web URL via HTTP (marked with a red circle labeled "2")
- Data source URL:** <https://vincentarelbundock.github.io/Rdatasets/csv/datasets/AirPassengers.csv> (marked with a red circle labeled "3")
- Data format:** CSV (marked with a red circle labeled "4")
- CSV or TSV has header row
- Use cached results

Execution details are listed below:

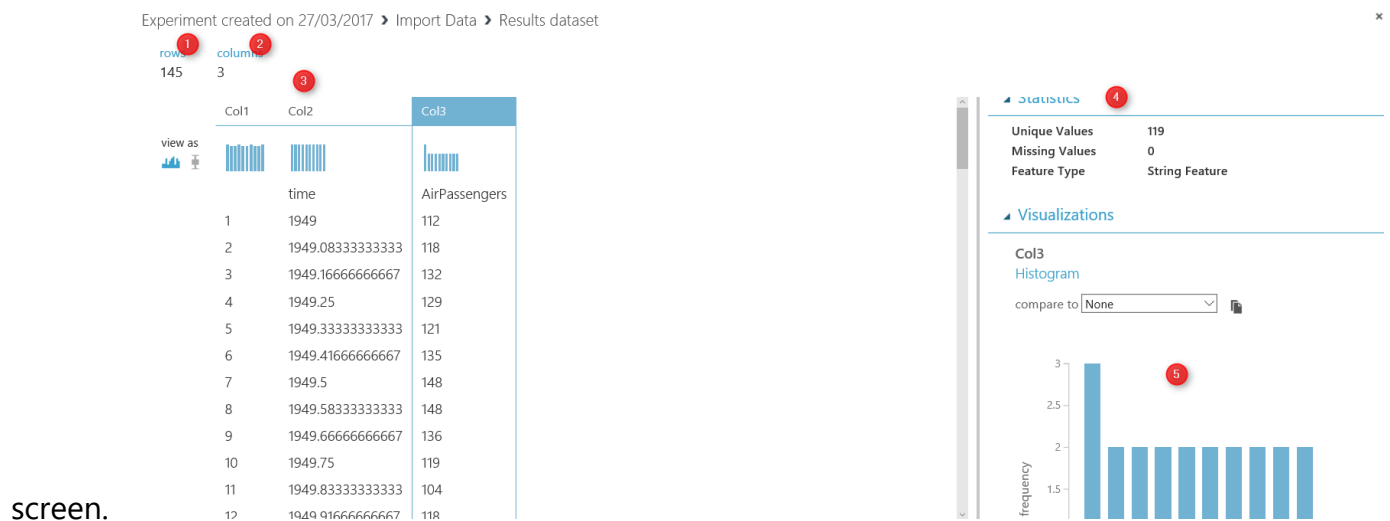
START TIME	3/27/2017 3:51:01 PM
END TIME	3/27/2017 3:51:10 PM
ELAPSED TIME	0:00:09.285
STATUS CODE	Finished
STATUS DETAILS	None

A "View output log" link is located at the bottom of the properties pane.

After running the experiment, you will see a correct green sign will appear in the front of the module, which shows that we are correctly able to import data from the web. Now by clicking on the node at the bottom of the module (number 1) and then clicking on the "Visualize" option (number 2) you will see the imported dataset.



The imported dataset has 145 rows (number 1) and three columns (number 2). Also, the detail of the data has been shown in the middle of the screen. Finally, a description of data both from statistic view(number 4) and from visual aspect(number 5) are shown in the right side of the



Experiment created on 27/03/2017 > Import Data > Results dataset

row	Col1	Col2	Col3
1	1949	time	112
2	1949.083333333333		118
3	1949.166666666667		132
4	1949.25		129
5	1949.333333333333		121
6	1949.416666666667		135
7	1949.5		148
8	1949.583333333333		148
9	1949.666666666667		136
10	1949.75		119
11	1949.833333333333		104
12	1949.916666666667		118

view as

Statistics

- Unique Values: 119
- Missing Values: 0
- Feature Type: String Feature

Visualizations

Col3

Histogram

compare to: None

frequency

screen.

I thought these are the main approaches for importing data into Azure ML. According to the machine learning process (Part1) after identifying the source data there is a need to do data wrangling in Azure ML. Hence, in the next Chapter I will explain how to use Azure ML different components for data wrangling.

---

---

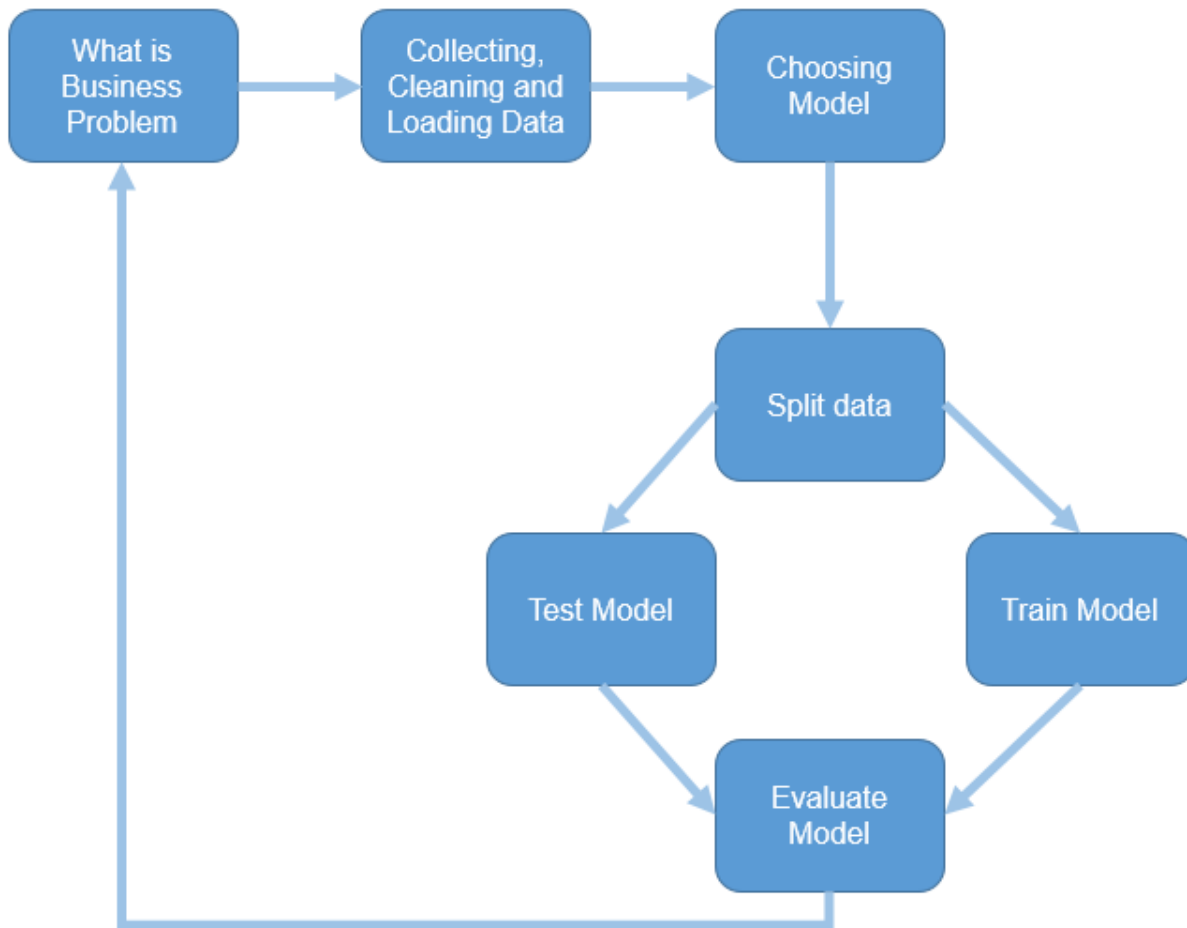
## Chapter 3: Data Transformation in R

Published Date: April 6, 2017



In a previous chapter I have explained how to import data into Azure ML Studio environment. In this part, I will show how to do data cleaning, data transformation in Azure ML Studio environment.

The second step in the machine learning process is about collecting (Chapter2), cleaning and loading data (current part).



Azure ML Studio has different components for data transformation (see below image ).

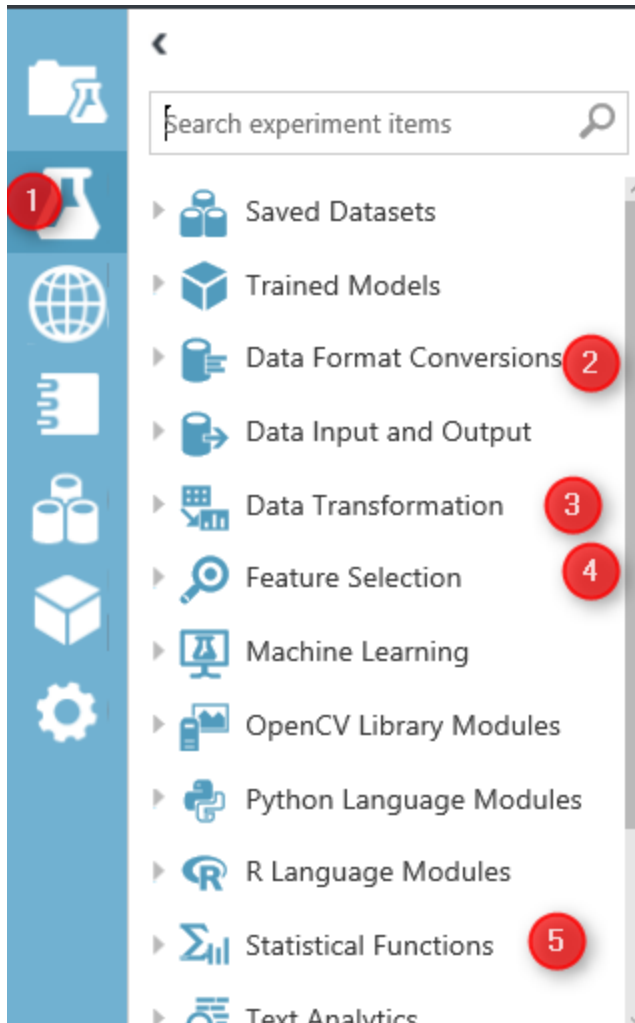
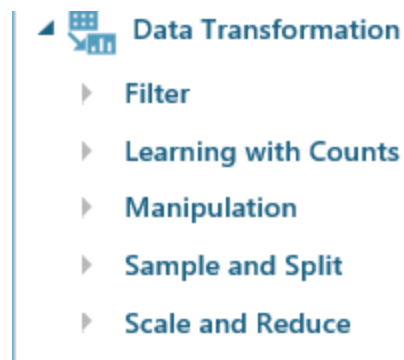


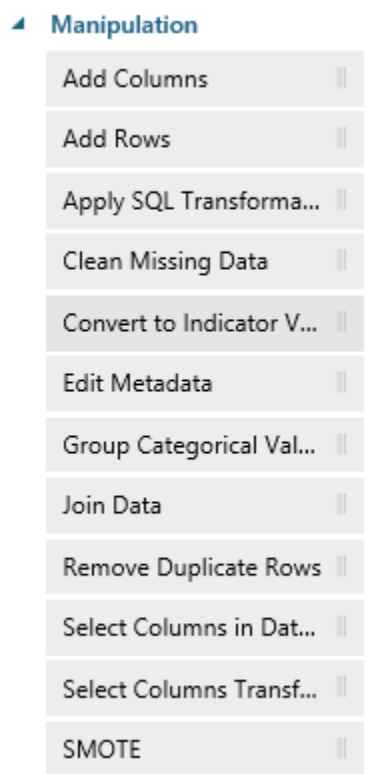
Image 1. Azure ML Studio Data Manipulation and Transformation.

In Azure ML Studio Studio, click on the Azure Experiment component (image1.1). There is a different type of components that can be used for the different scenario of data transformation.

In this chapter I will talk about “Data Transformation” (Image 1.3) component and in detail data manipulation. This component has many things that help us to transform data, clean data and so forth.

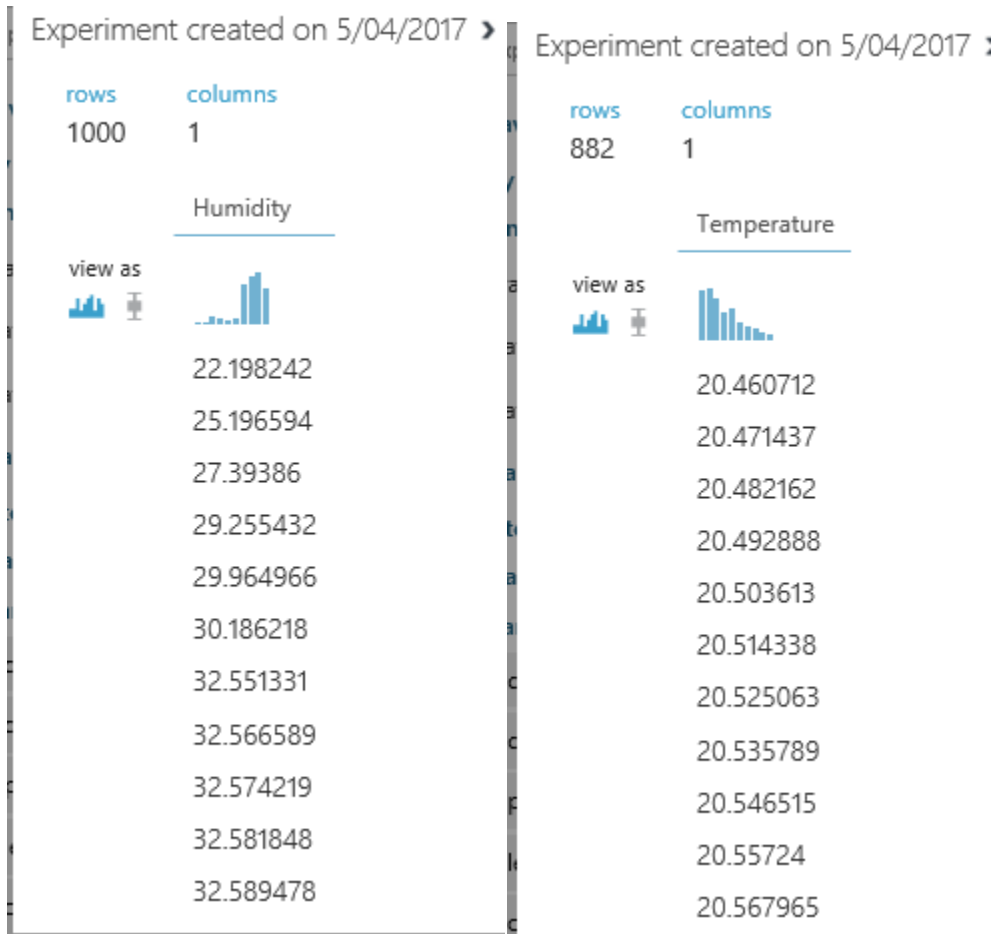


the main Chapter that can be used in most of Machine Learning experiments is "Manipulation" . in the below image you will be able to see the different components



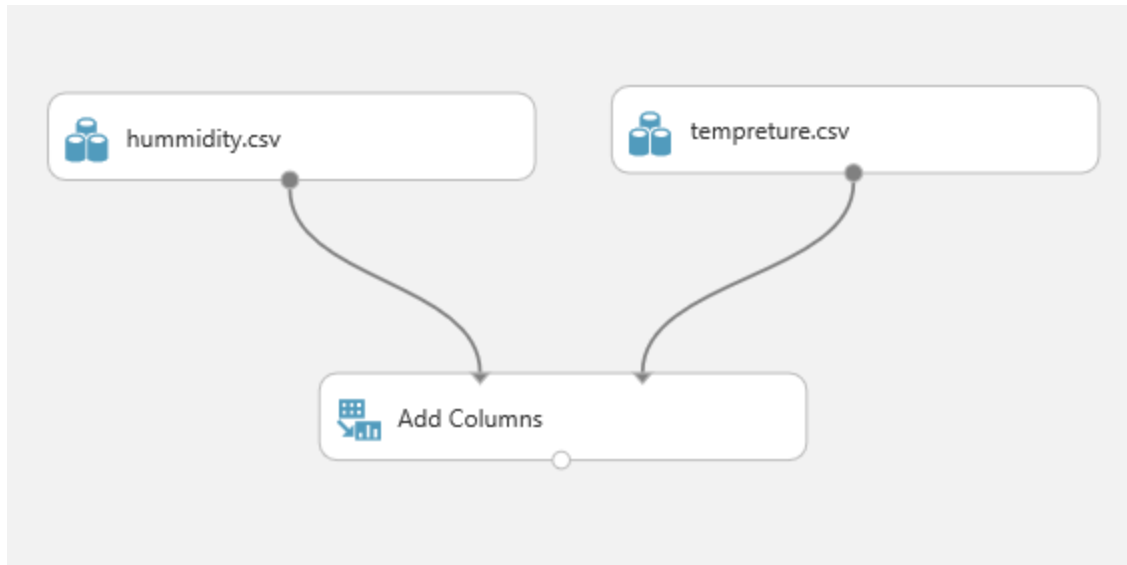
## Add Columns

Add Columns is one of the primary types of data transformation that helps you to combine two datasets. For example, imagine we have a dataset about weather Humidity and another one about the weather temperature.



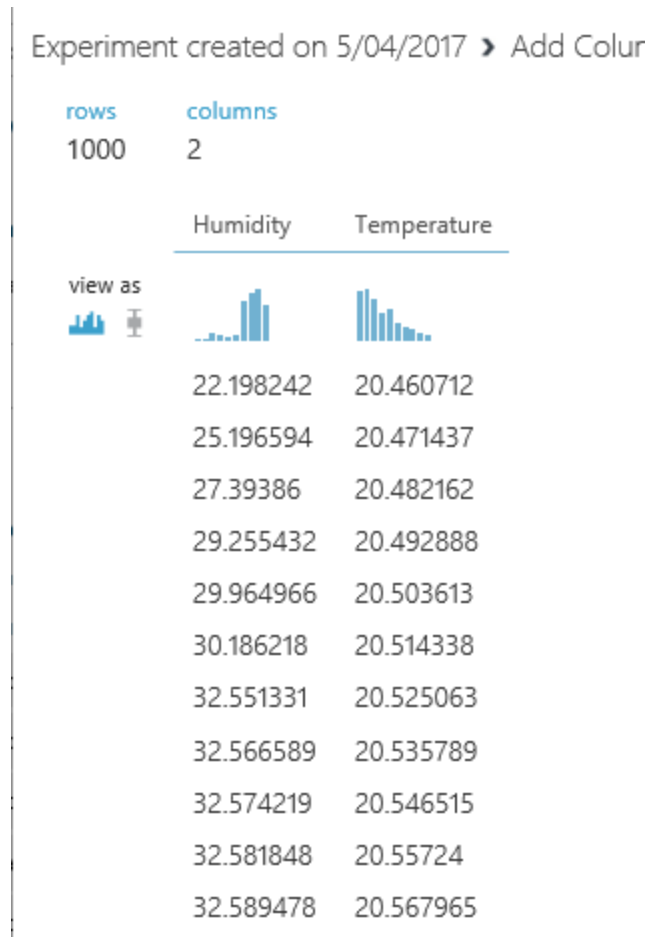
there are in different data set and you want to combine them.





The first step is to drag and drop these two data set into the experiment area. Then connect them to the "Add Column" module (see above image).

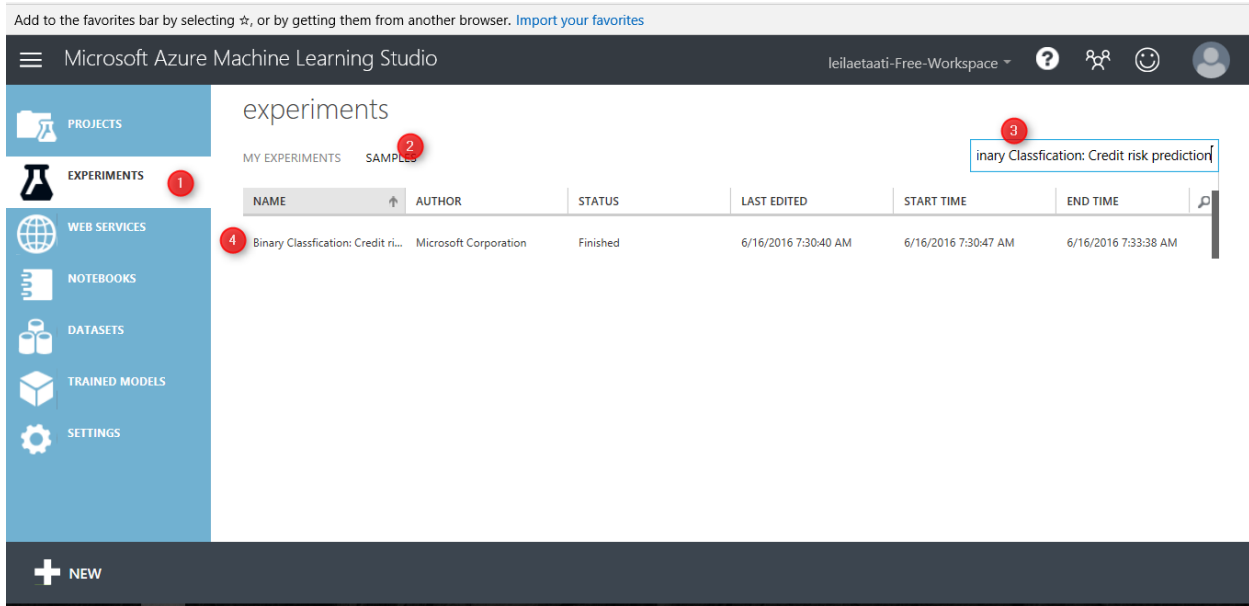
if right click on the node bottom of the module "Add Columns" you will see that we have a dataset that has both Temperature and Humidity data in one place.



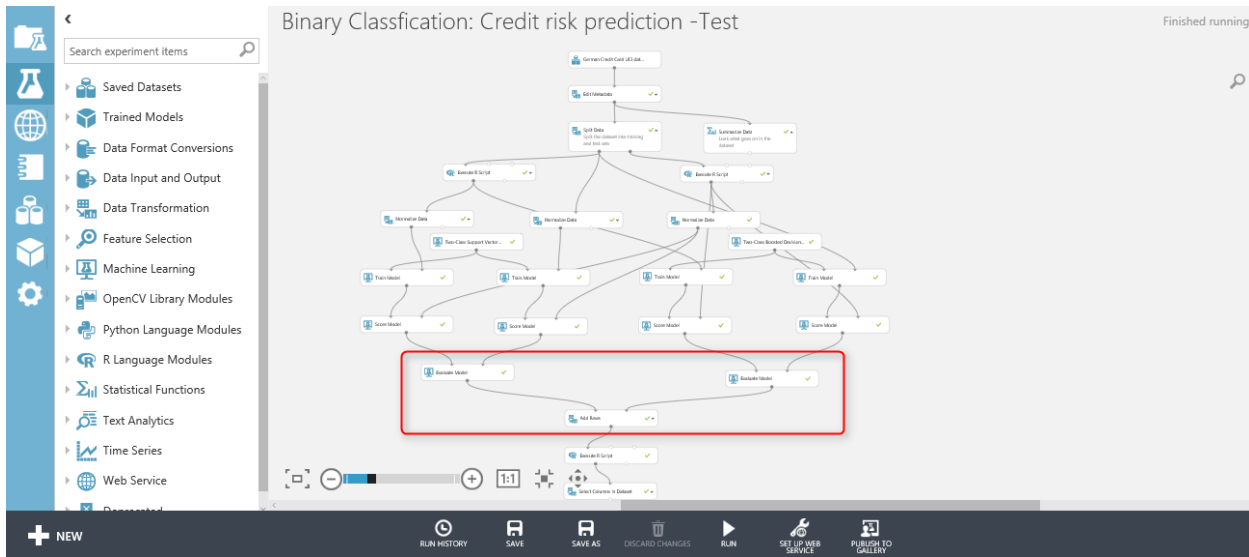
### Add rows

This component helps us to add combine rows that we have in a dataset or combine the evaluation results. Imagine that we created an experiment that runs two machine learning, we want to compare and show the result of the evaluation for each of them.

I am going to explain this component via an existing Experiment in Sample experiment in Azure ML Studio. To start, click on the experiment option (below image), then click on the sample option (number 2), then in the search area type the "Binary Classification: Credit risk prediction" and click on the sample to open it.



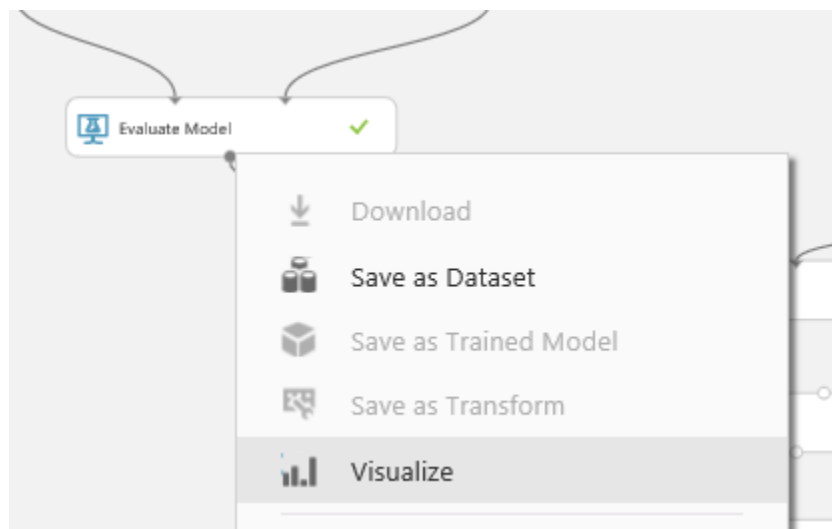
when you open the sample do not freak out! I will talk about all other components in this experiment later in Machine learning parts. Just click on the Run at the bottom of the page and look at the "Add rows" and evaluation models' components.



as you can see in the below image, we have two "Evaluate model" components (will talk about it in Machine learning part).



First, visualize the “Evaluation Model” component by right-clicking on the node at the bottom of the component (See below image).



Then click on the “Visualize” option (above image). And you will see the evaluation result page. Just scroll down to reach the next part of the evaluation result as below. This component shows the accuracy of the model to us with charts and numbers. However, in add rows component only the numbers and data will be shown.

Binary Classification: Credit risk prediction -Test

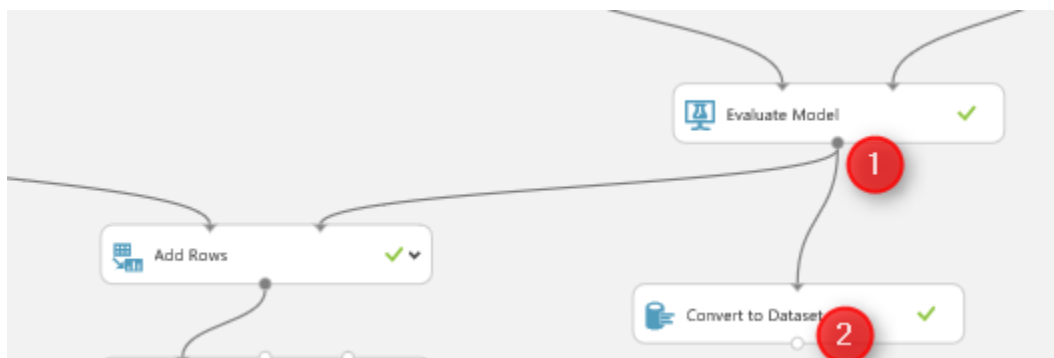
Binary Classification: Credit risk prediction -Test > Evaluate Model > Evaluation results

True Positive	False Negative	Accuracy	Precision	Threshold	AUC
455	295	0.659	0.850	0.5	0.749
False Positive	True Negative	Recall	F1 Score		
80	270	0.607	0.708		
Positive Label	Negative Label				
2	1				

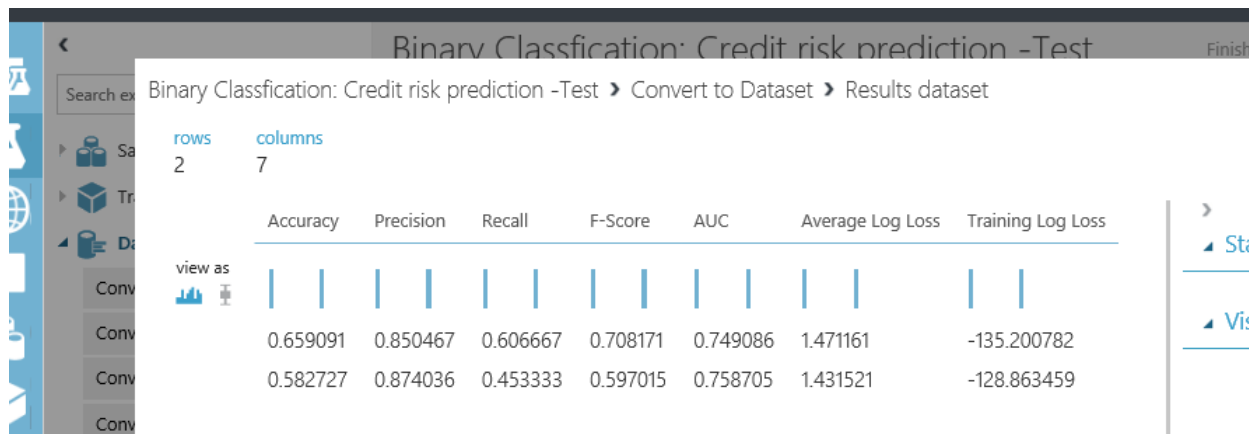
  

Score Bin	Positive Examples	Negative Examples	Fraction Above Threshold	Accuracy	F1 Score	Precision	Recall	Negative Precision	Negative Recall	Cumulative AUC
(0.900,1.000]	300	43	0.312	0.552	0.549	0.875	0.400	0.406	0.877	0.028
(0.800,0.900]	60	12	0.377	0.595	0.618	0.867	0.480	0.431	0.843	0.044
(0.700,0.800]	40	10	0.423	0.623	0.658	0.860	0.533	0.449	0.814	0.058
(0.600,0.700]	35	9	0.463	0.646	0.691	0.855	0.580	0.467	0.789	0.072
(0.500,0.600]	20	6	0.486	0.659	0.708	0.850	0.607	0.478	0.771	0.082
(0.400,0.500]	10	5	0.500	0.664	0.715	0.845	0.620	0.482	0.757	0.091
(0.300,0.400]	20	8	0.525	0.675	0.730	0.839	0.647	0.492	0.734	0.106
(0.200,0.300]	15	9	0.547	0.680	0.740	0.831	0.667	0.498	0.709	0.123
(0.100,0.200]	30	17	0.590	0.692	0.758	0.817	0.707	0.512	0.660	0.156
(0.000,0.100]	220	231	1.000	0.682	0.811	0.682	1.000	1.000	0.000	0.749

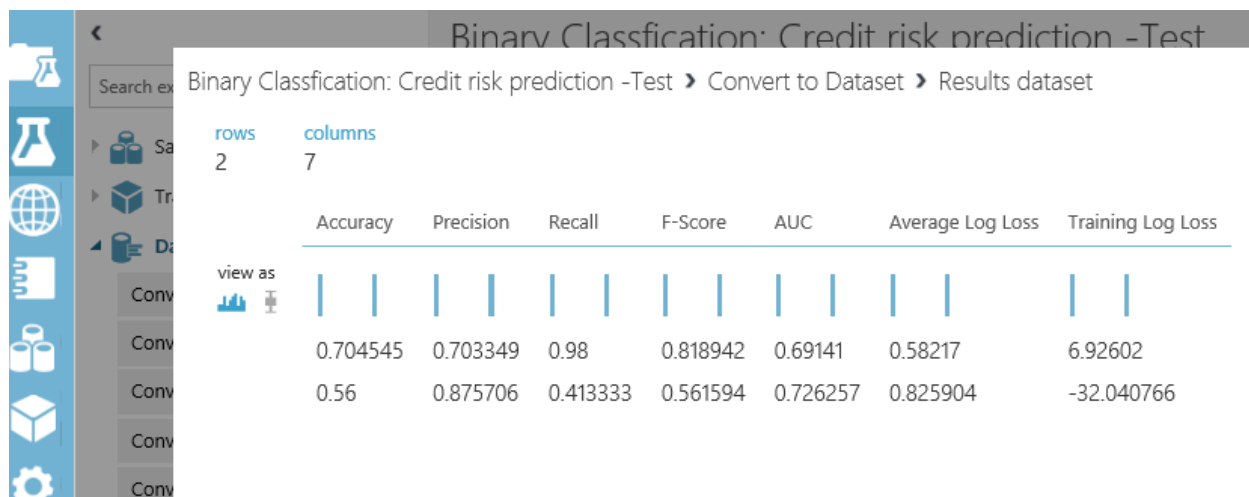
However, the node "Evaluation Model" has some also data output, to see just data not the graphs, I am going to drag and drop another component to model name as "Convert to Dataset" as below. I just did this to show you the real data that will pass to the "Add Rows" component.



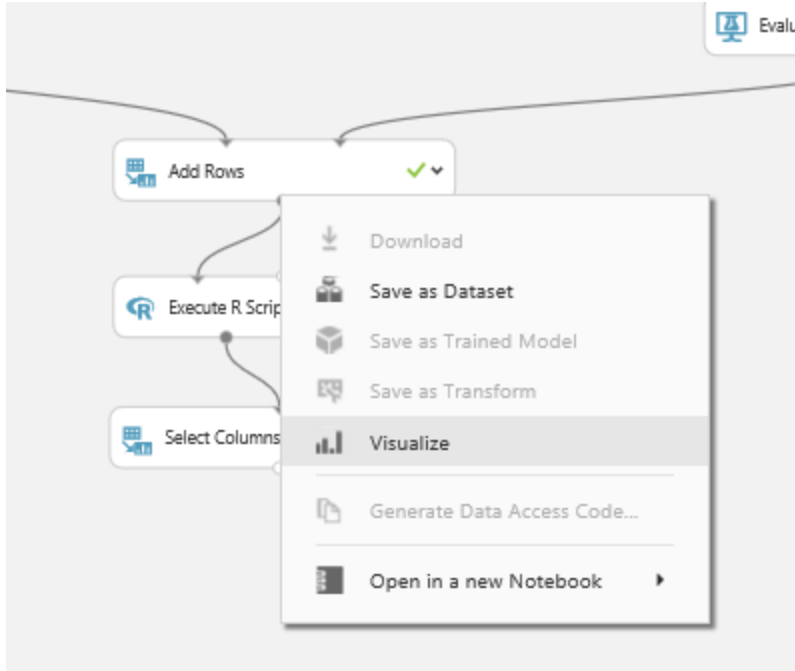
just connect it to the evaluation model and then right click on the bottom of "Convert to Dataset" component to visualize the data (see below picture)



we have the same for other "evaluation model" component as below



"Add Row" component will merge this information. As you can see in the above picture, the raw data of the "evaluation model" is just two rows with seven columns. That each of them tries to explain the accuracy of the model to end user. We want to compare the results of these two algorithms with other ones. Hence, to see the results, right click on the bottom of the "Add Rows" component, then click on the "Visualize" option.



the below picture was shown to you. as you can see "Add Rows" has merged these data in one place

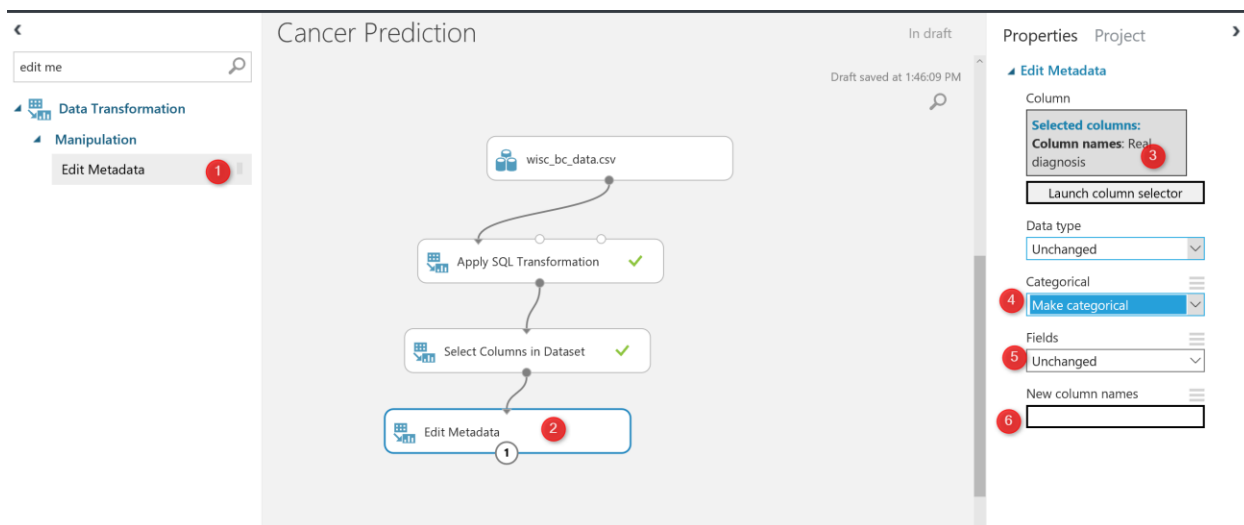


and we able to compare the result together. I will talk about the evaluation model later in Machine learning part.

Next Chapter, I will talk about the other component for data transformation.

# Chapter 4: A Machine Learning Prediction Scenario – Data Cleaning

Published Date: June 1, 2017

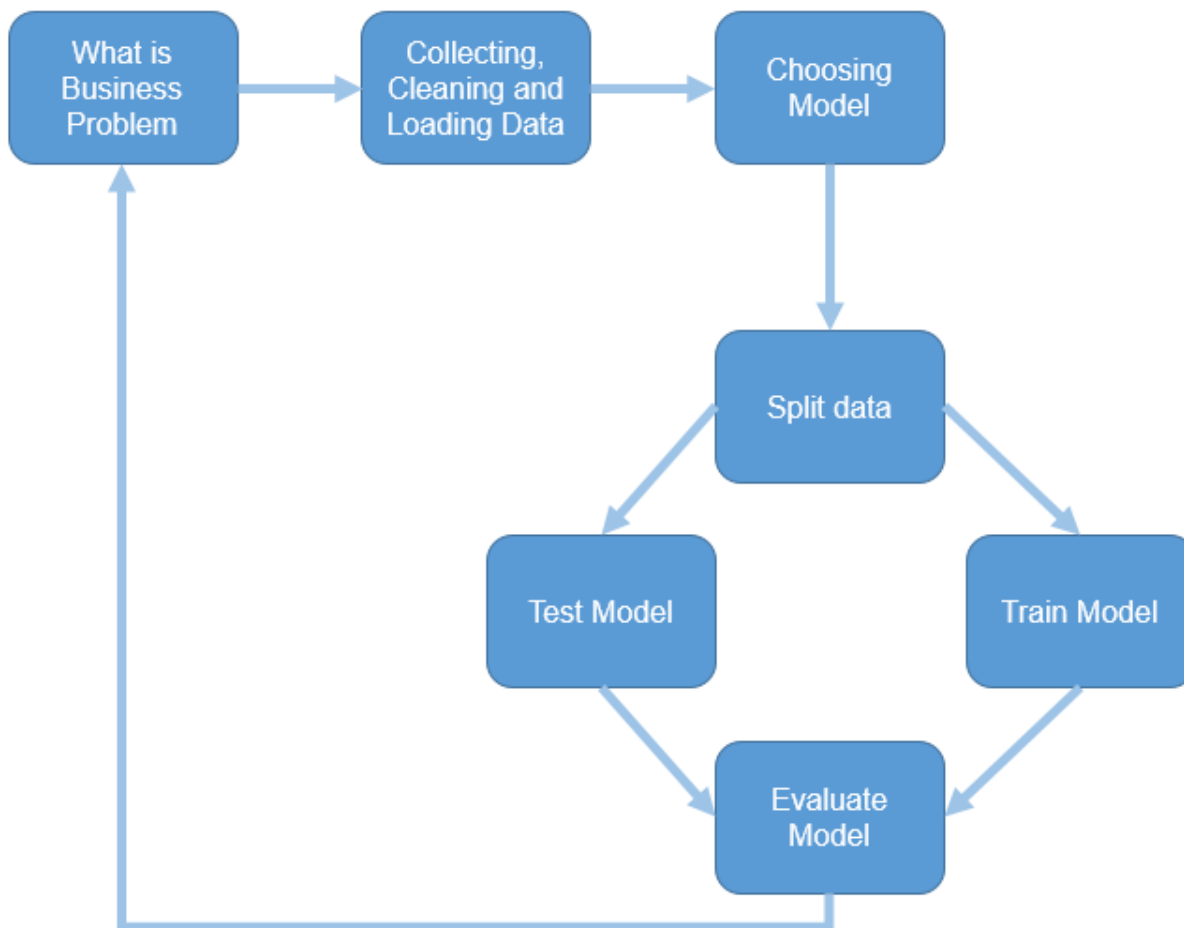


In previous Chapters Chapter1, Chapter 2 and Chapter 3 I have explained some about the Azure ML Studio environment, how to import data into it and finally how to do data transformation using Azure ML Studio component.

In this chapter and the next one I am going to show how to do a Machine Learning in Azure ML Studio using different components via a scenario.

First based on the Machine Learning Process (see below image), the first step is to identify the business problems

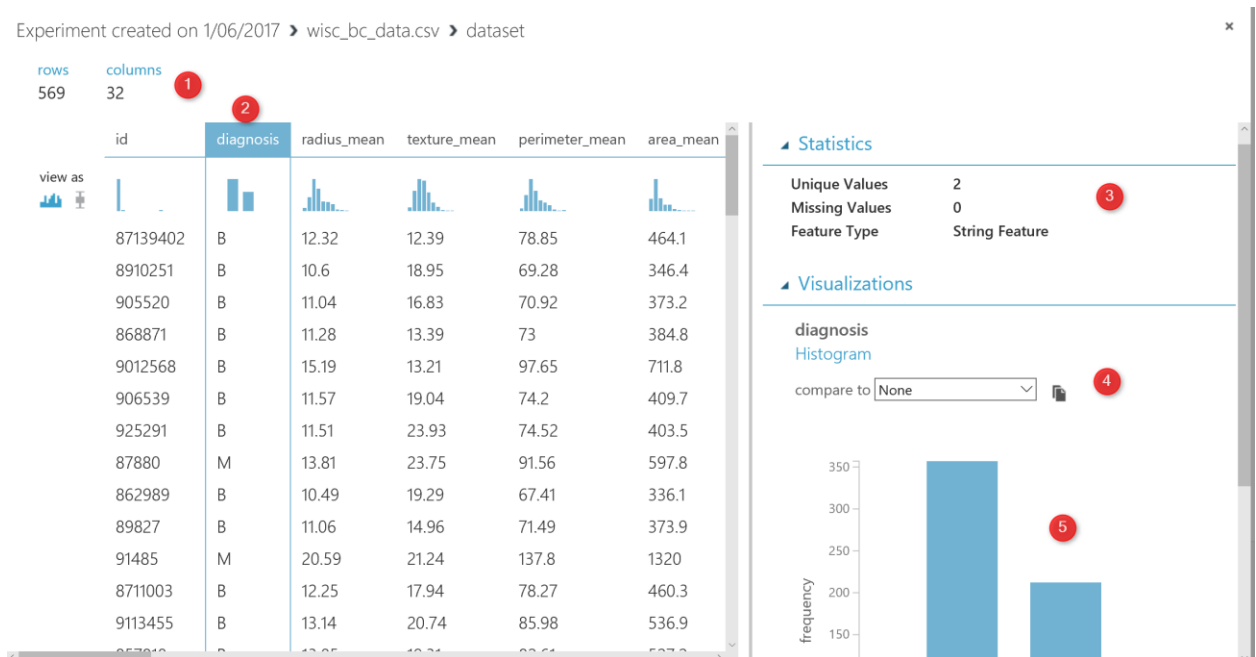
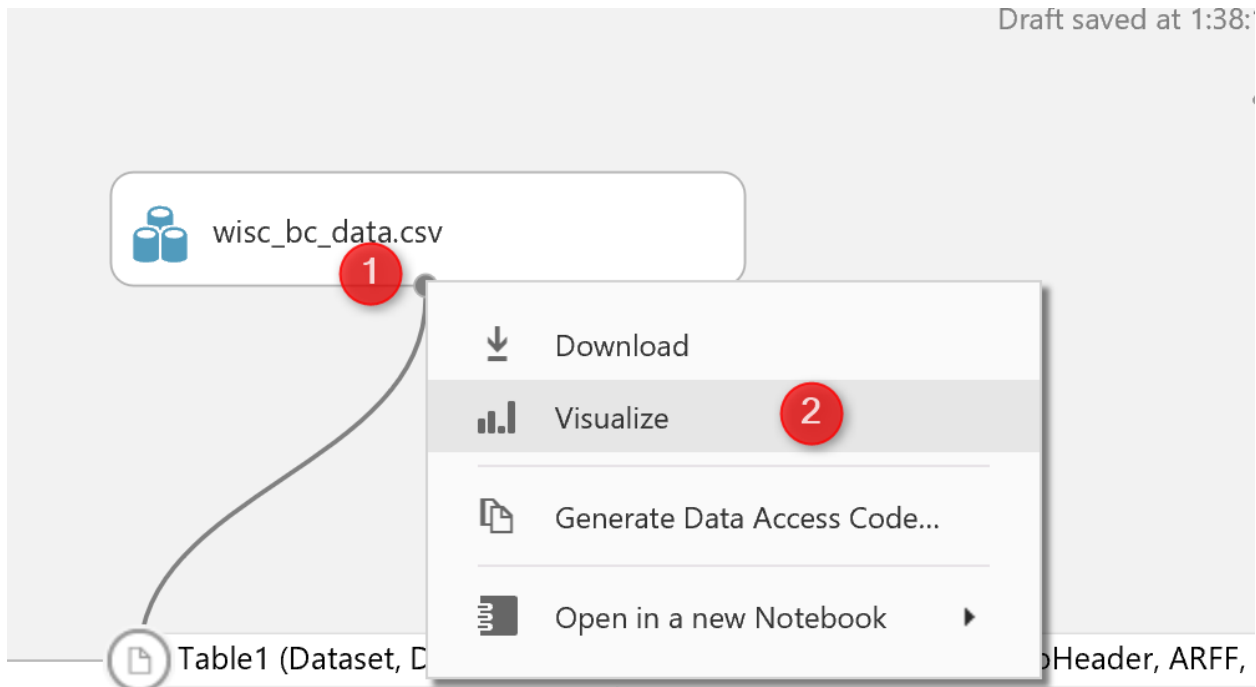




### What is the Business Problem?

I am going to predict whether a customer will be **Benign or Malignant**. I have a dataset that shows the laboratory results of cancer cells, also the patients ID, and the final doctor 's diagnosis as "B" and "M".

I have imported the data as a "CSV" file, the CSV file has **569 records** and **32 columns** (see below picture). In the right side of the picture, you will see a menu that shows the statistics about the data (number 3). Moreover, in number 4 and 5, you able to see the chart that shows the histogram of the diagnosis column data.

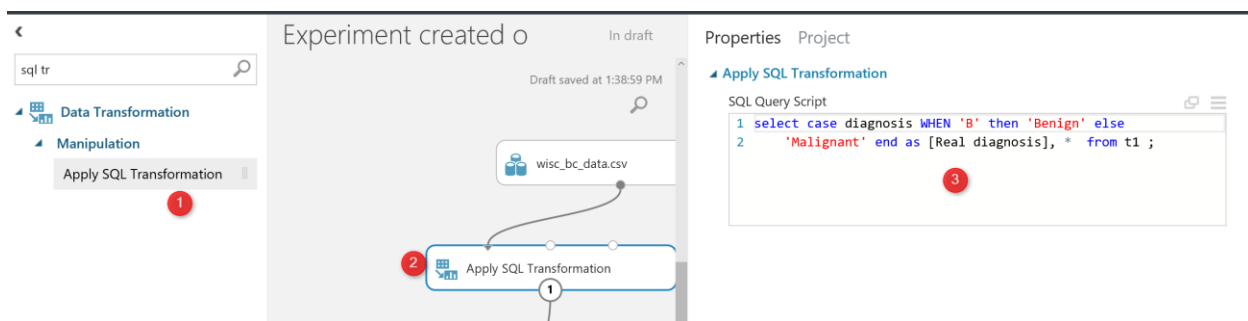


I am going to change the data description of the diagnosis. I want to replace "B" with "Benign" and "M" with "Malignant". This can be done using "SQL Transformation". SQL Transformation component can be accessed under the "Data Transformation" component. Just drag and drop it

to the experiment area, then connect it to the dataset as number 2 in the below picture. To write SQL code click on the component, then write a normal SQL Statement to transfer data as below

```
“select case diagnosis WHEN ‘B’ then ‘Benign’ else
‘Malignant’ end as [Real diagnosis], * from t1;”
```

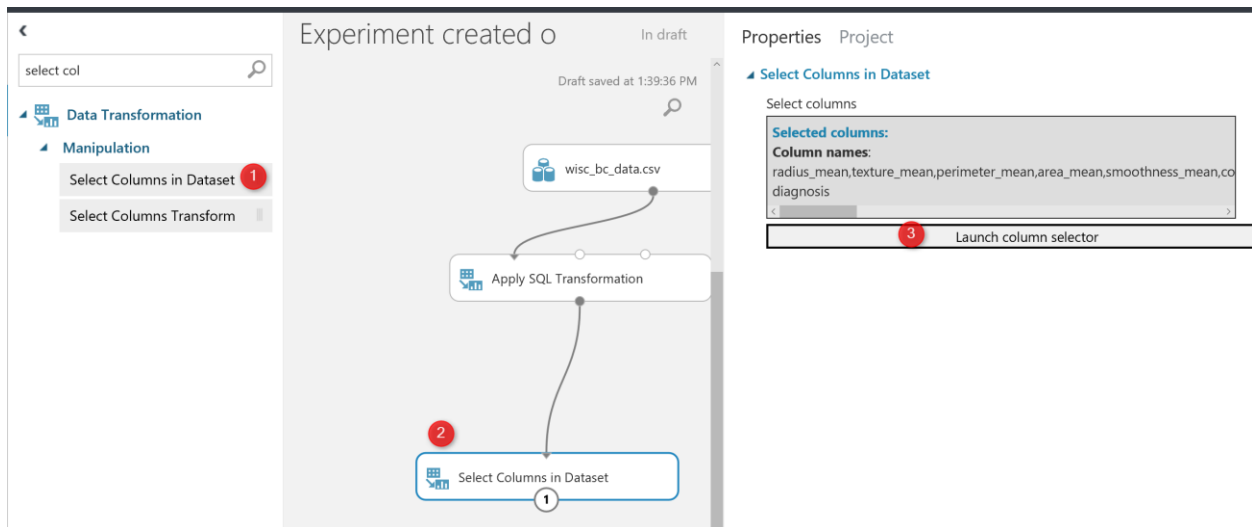
The code will replace the value of column “diagnosis”. as you can see in the above code, the data has been selected from the “t1” table. So, what is “T1” table, as you can see in the below picture (number 2). SQL transformation component has three main input nodes. The first node is called “t1” as we connect the dataset to it. so, if you connect your dataset to the second node then in SQL scripts you should select your data from “t2”



The next step for **Data cleaning** is to remove the columns that we think does not have an impact on the prediction.

### Select Column Data

I am going to remove the “**Patient ID**” from the dataset using “**Select Columns in Dataset**” to select the columns that I need.



after running the code, we will see the below results, so the "ID" columns have been removed and we have now 31 columns instead.

Experiment created on 1/06/2017 > Select Columns in Dataset > Results dataset

rows: 569, columns: 31

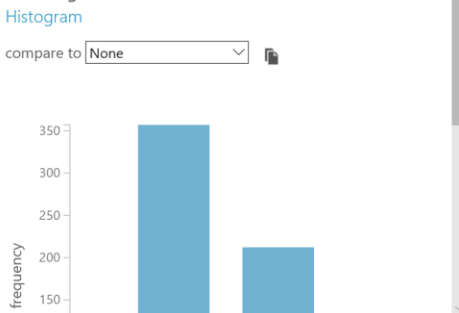
Real diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness
Benign	12.32	12.39	78.85	464.1	0.1028
Benign	10.6	18.95	69.28	346.4	0.09688
Benign	11.04	16.83	70.92	373.2	0.1077
Benign	11.28	13.39	73	384.8	0.1164
Benign	15.19	13.21	97.65	711.8	0.07963
Benign	11.57	19.04	74.2	409.7	0.08546
Benign	11.51	23.93	74.52	403.5	0.09261
Malignant	13.81	23.75	91.56	597.8	0.1323
Benign	10.49	19.29	67.41	336.1	0.09989
Benign	11.06	14.96	71.49	373.9	0.1033
Malignant	20.59	21.24	137.8	1320	0.1085
Benign	12.25	17.94	78.27	460.3	0.08654
Benign	13.14	20.74	85.98	536.9	0.08675

Statistics:

- Unique Values: 2
- Missing Values: 0
- Feature Type: String Feature

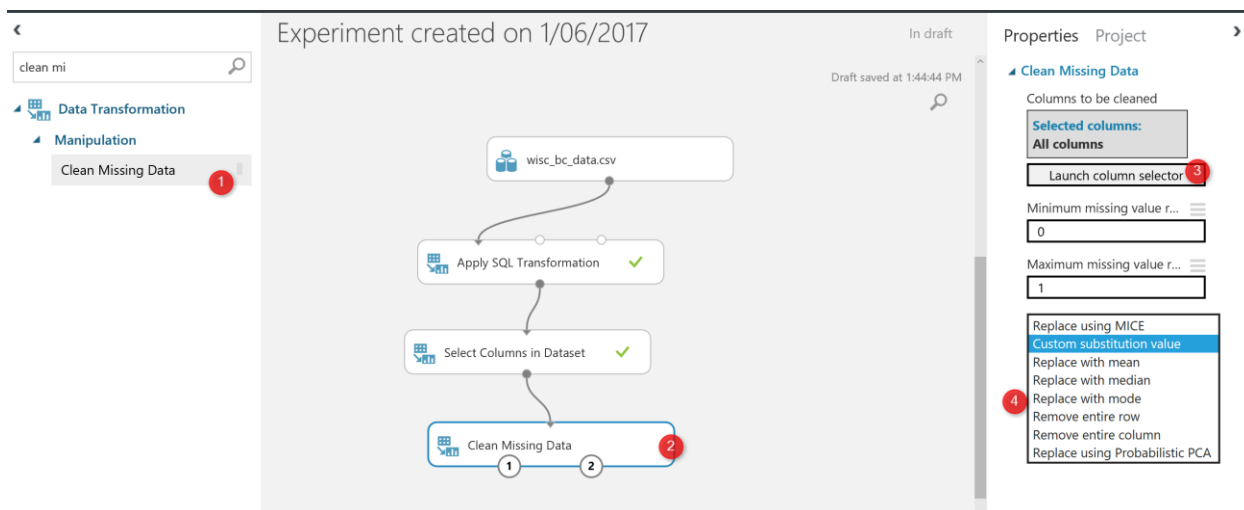
Visualizations:

Real diagnosis Histogram

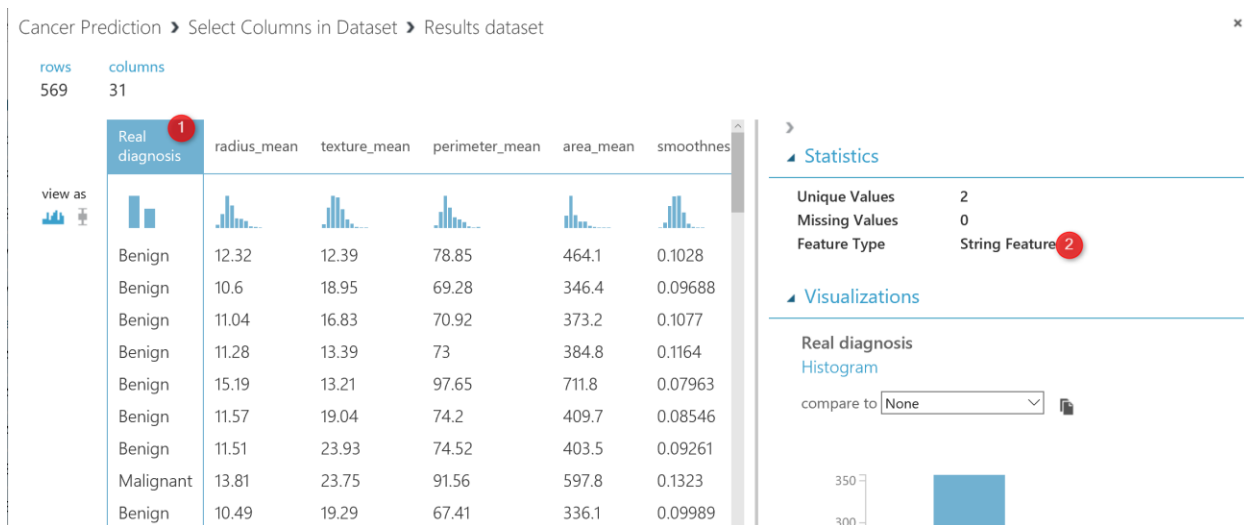


There is some possibility to have "Missing Values". The missing value may impact on the prediction results, so always recommend to remove them. There is a component in Azure ML

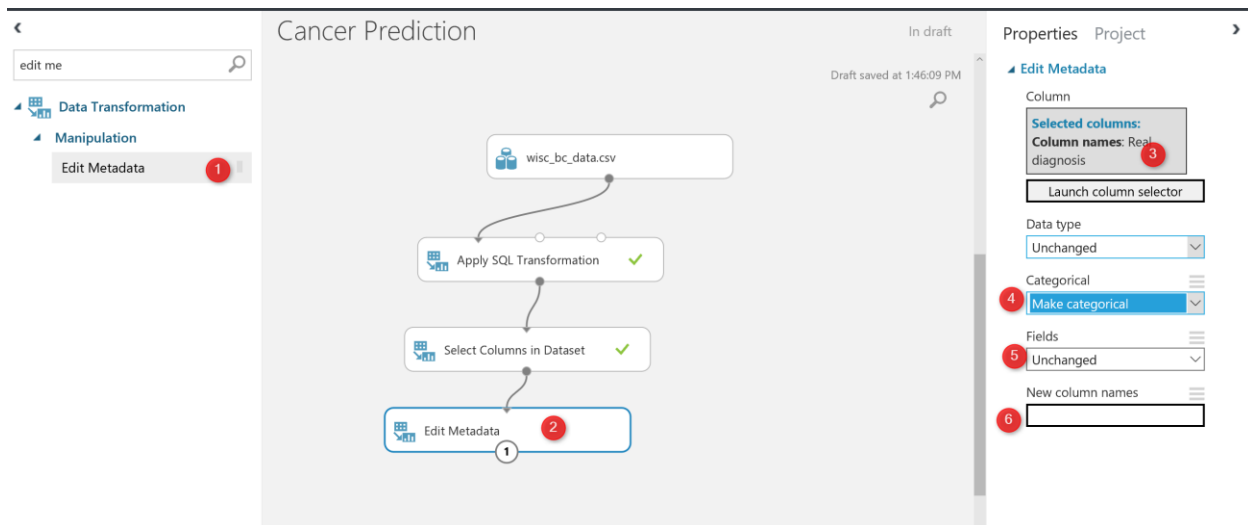
Studio called **“Clean Missing Values”** as shown below. This component can remove data that does not have a value. (see below picture).



The next step for data cleaning is to change the data structure, the diagnosis data column was **“string”** type ( see the below picture ), before adding the edit metadata component, the diagnosis column was **“String Feature”**.



for making the prediction, we are going to use **“Two-Class”** algorithms that need a **“categorical”** data, hence we need to change the diagnosis data type. To change the data type in Azure ML Studio we have a component that helps us name **“Edit Metadata”**.



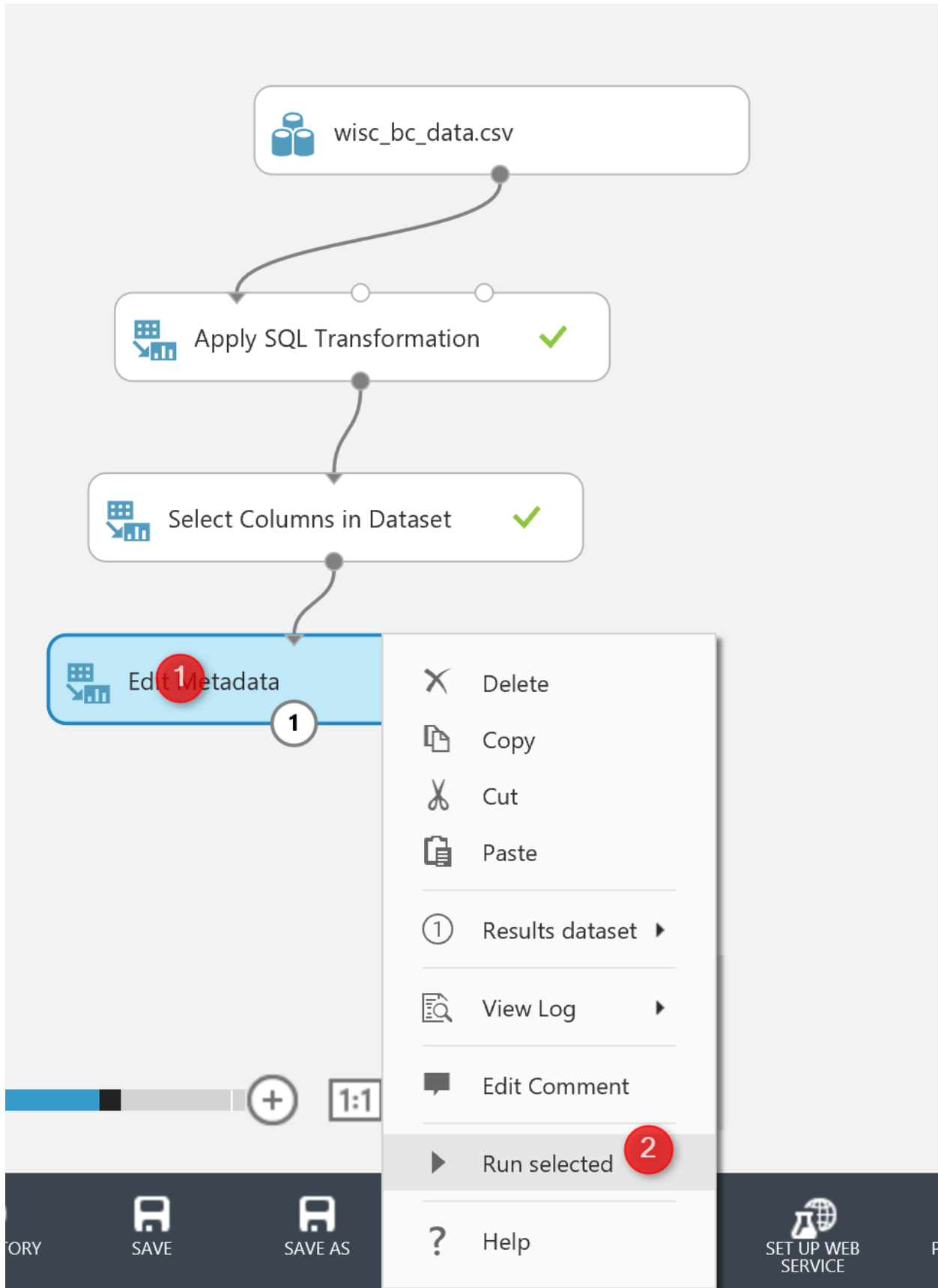
As you can see in the above picture, I have drag and drop this component into the experiment area, then in number 3, I specify the column that I need to change the data structure. Following, in number 4, I change the column to **categorical data**.

You are also able to change the name of the column in the Edit metadata component as shown in number 6.

Now I am going to run the code. However I want to see the final result of adding "Edit Metadata".

There is a possibility to run just one node at the time in Azure ML Studio to make experiment creation faster.

For instance, after adding the "**Edit Metadata**" I just right-click on the node and I have selected the "Run selected node".



The screenshot displays a workflow in Azure Machine Learning Studio. The workflow consists of the following steps:

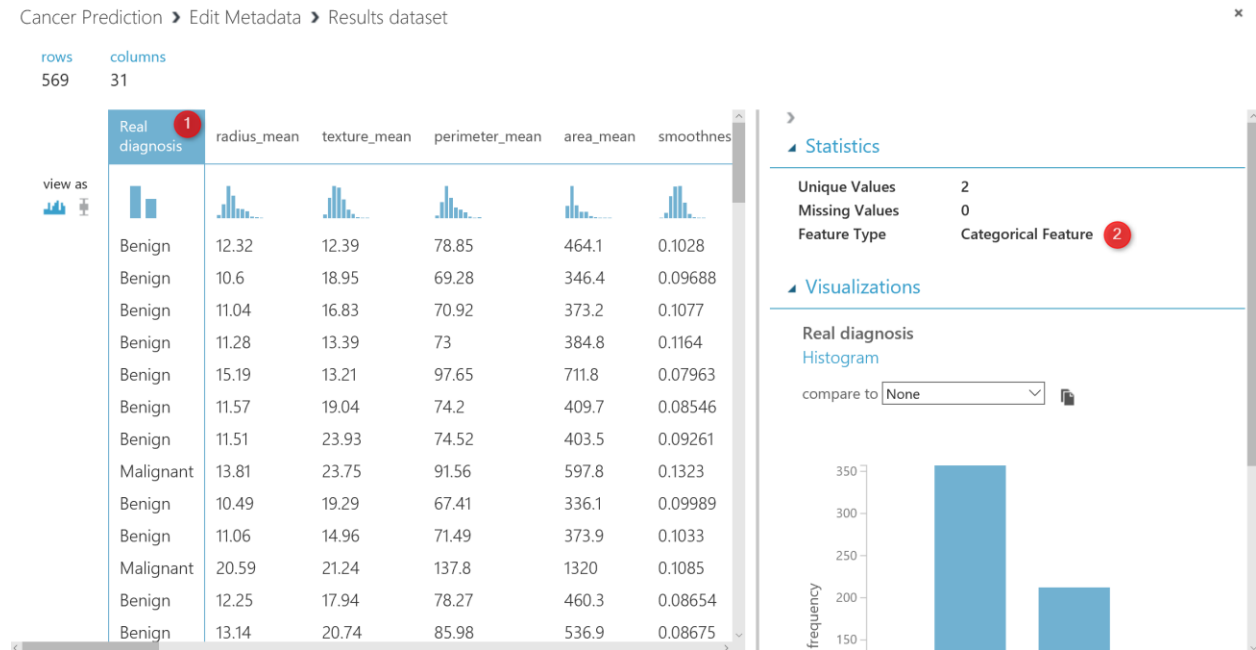
- wisc\_bc\_data.csv**: The initial data source.
- Apply SQL Transformation**: A step that has been completed, indicated by a green checkmark.
- Select Columns in Dataset**: A step that has also been completed, indicated by a green checkmark.
- Edit Metadata**: The current step, highlighted in blue. A red circle with the number '1' is placed over the 'Edit Metadata' text.

A context menu is open over the 'Edit Metadata' step. The menu items are:

- Delete
- Copy
- Cut
- Paste
- Results dataset (with a right-pointing arrow)
- View Log (with a right-pointing arrow)
- Edit Comment
- Run selected** (highlighted in grey, with a red circle containing the number '2' next to it)
- Help

At the bottom of the interface, there is a toolbar with icons for 'COPY', 'SAVE', 'SAVE AS', and 'SET UP WEB SERVICE'. A zoom level indicator shows '1:1'.

Now If I visualized the data at the node “ **Edit Meta Data**” you will see that the diagnosis column has a “ **categorical Feature**” data type which good for making the prediction.



there are still some steps for doing the data cleaning and feature election, that I will explain in the next Chapter.

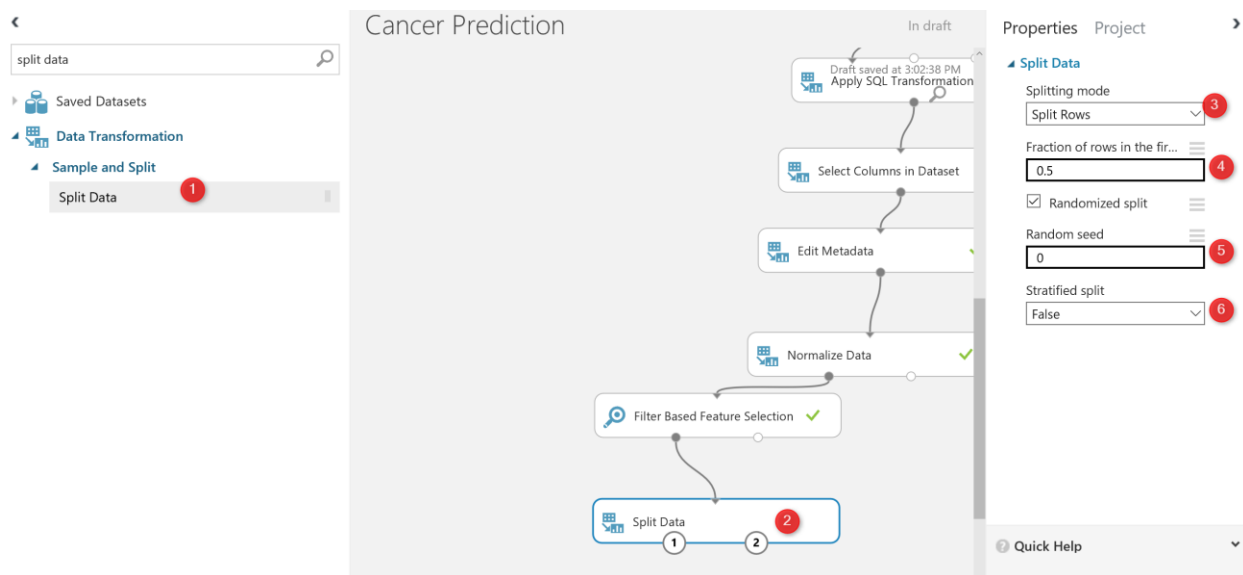
To sum up, in this chapter, I have explained how to work with “**SQL Transformation**” Component, “**Select specific columns**” to removing attributes. How to remove missing value using “**Clean Missing Data**” component. Also edit metadata to change the name and type of columns.

I next chapter I will explain how to **normalize** these data, how to find which attributes have more impacts on the “diagnosis” columns prediction, also how to **split data, train model, score morel, evaluate the model**.



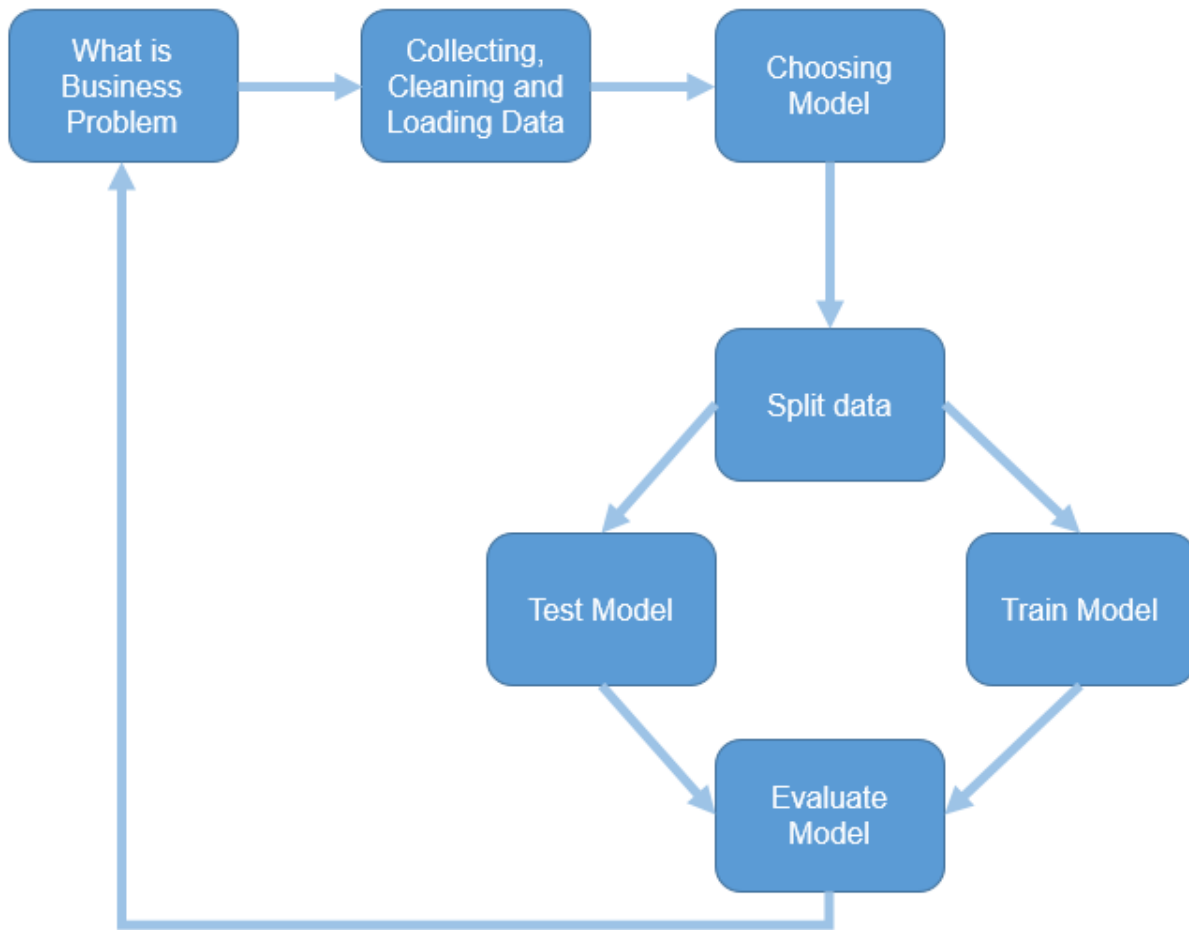
# Chapter 5: A Machine Learning Prediction Scenario – Feature Selection

Published Date: June 2, 2017

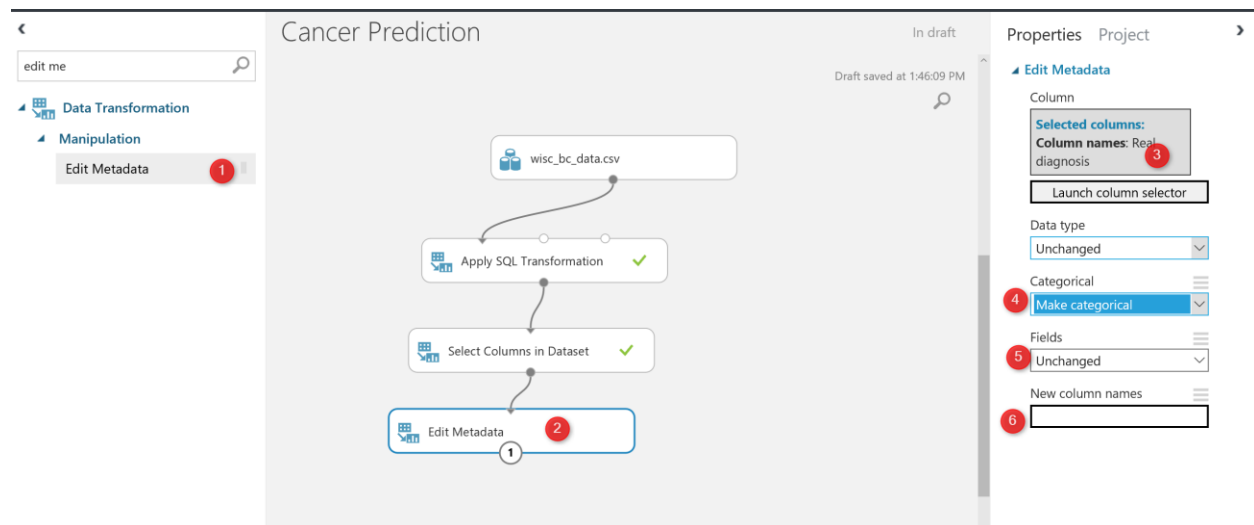


In the previous chapter, I start to do prediction the cancer diagnosis using some laboratory data. I have explained some of the main components for doing the data cleaning such as **“SQL Transformation”**, **“Edit Meta Data”**, **“Select Columns”** and **“Missing Values”**.

In this chapter, I am going to show the rest of the data cleaning process using Azure ML Studio components and how to split data for training.



In the last chapter, we come up with the below process.



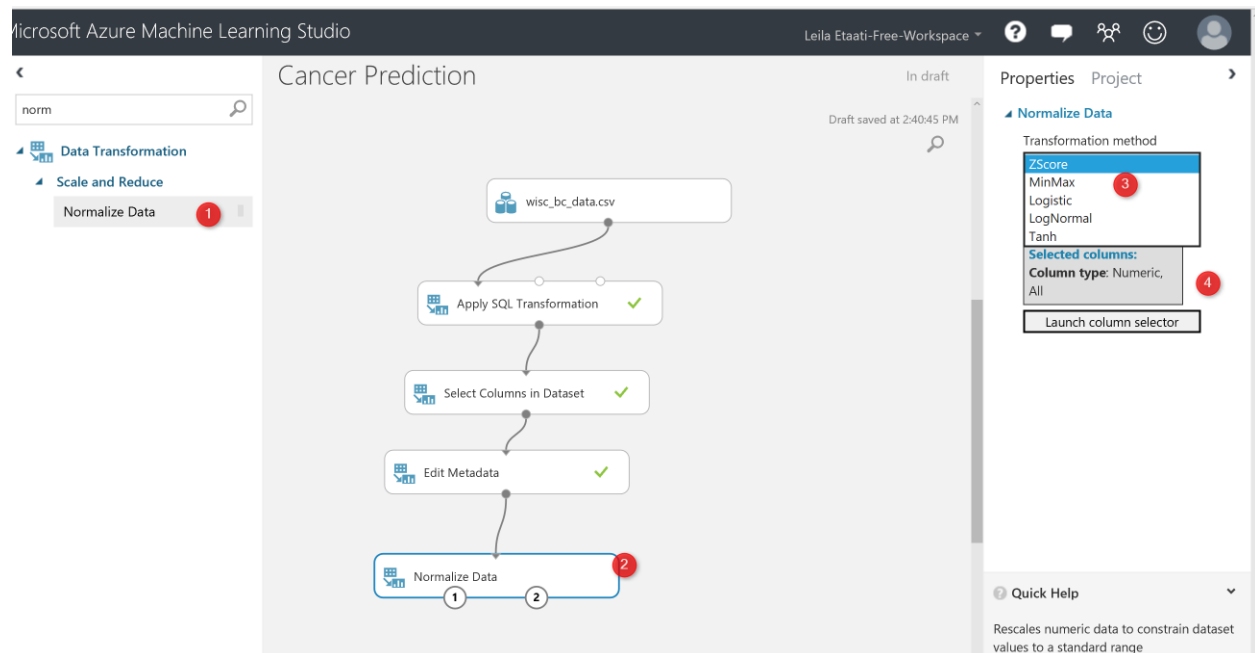
Now we are going to do some more data cleaning as “**Normalization of Data**”.

Look at the output of the data from “Edit metadata”:



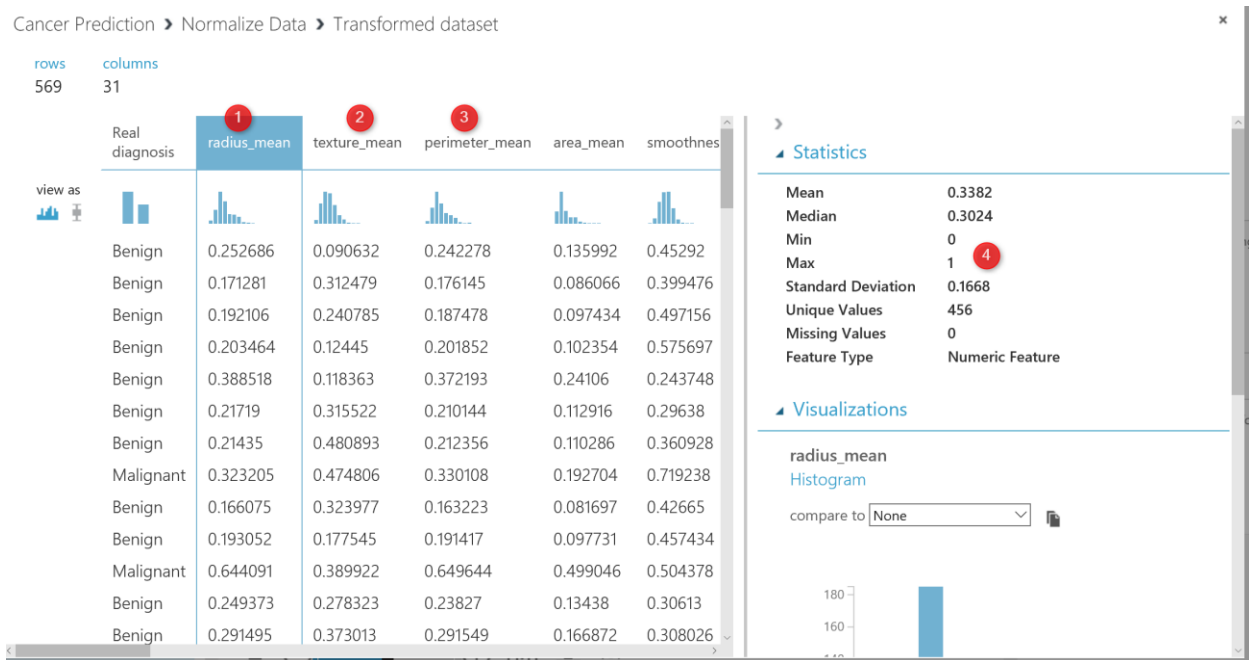
If you look at the data, you will see that each column has its data range, for instance column 2 (number 1 in the above picture ) has data range between 9 to 20, whilst the column number 5 (number 2 in picture) has values between 100 to 500, the same for column 6 (number 3), the data range is between 0.01 to 0.1. So the data is not in the same range. To do machine learning, it is important that all data be in the same range. I am going to bring data in a range of 0 to 1 using the “**Min-Max**” algorithm. There is a component in Azure ML Studio name “**Normalize Data**”.

As you can see in the below picture, Normalize data component exists under “**Data Transformation**” Component.



I drag and drop the component to the experiment area, in the right side of the experiment, we able to specify the normalization method (number 4 in the above picture), for this experiment I have to choose the “**Min-Max**” method. Also, we able to select which column we want to normalize (number 5 in the picture).

After running the experiment we will have below data set that is normalized in comparison with the previous one.



So, now we have enough data cleaning and data wrangling in our dataset. The next step is about the "Choosing the right data for prediction" that we call it "**Feature Selection**".

### Feature selection

Is the process of finding which attributes ha more impact on the prediction columns. In our example, we are going to see which laboratory measure has more impact on the Diagnosis result.

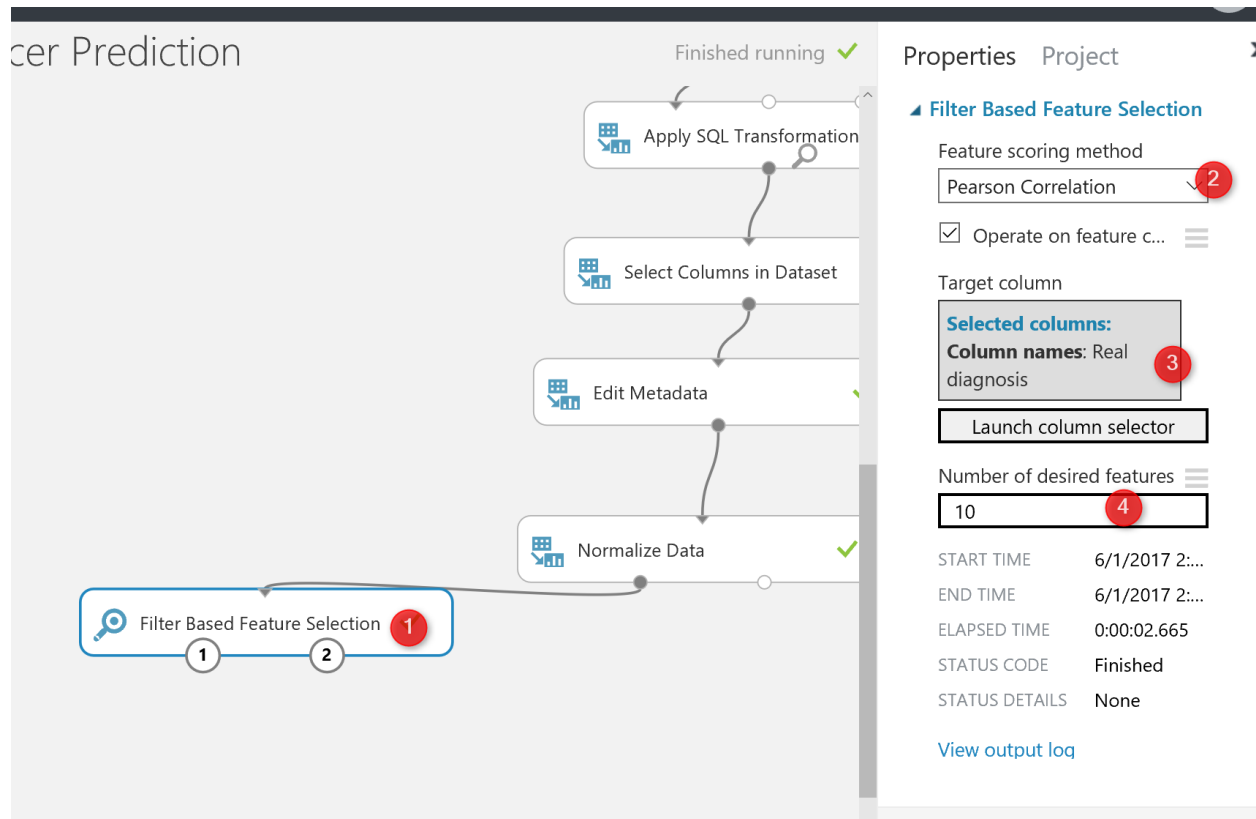
There are many approaches in machine learning to do that using algorithms like "**regression, decision tree**", **correlation analysis** will help.

In Azure ML Studio there is a component name "**Filter Based Feature Selection**".

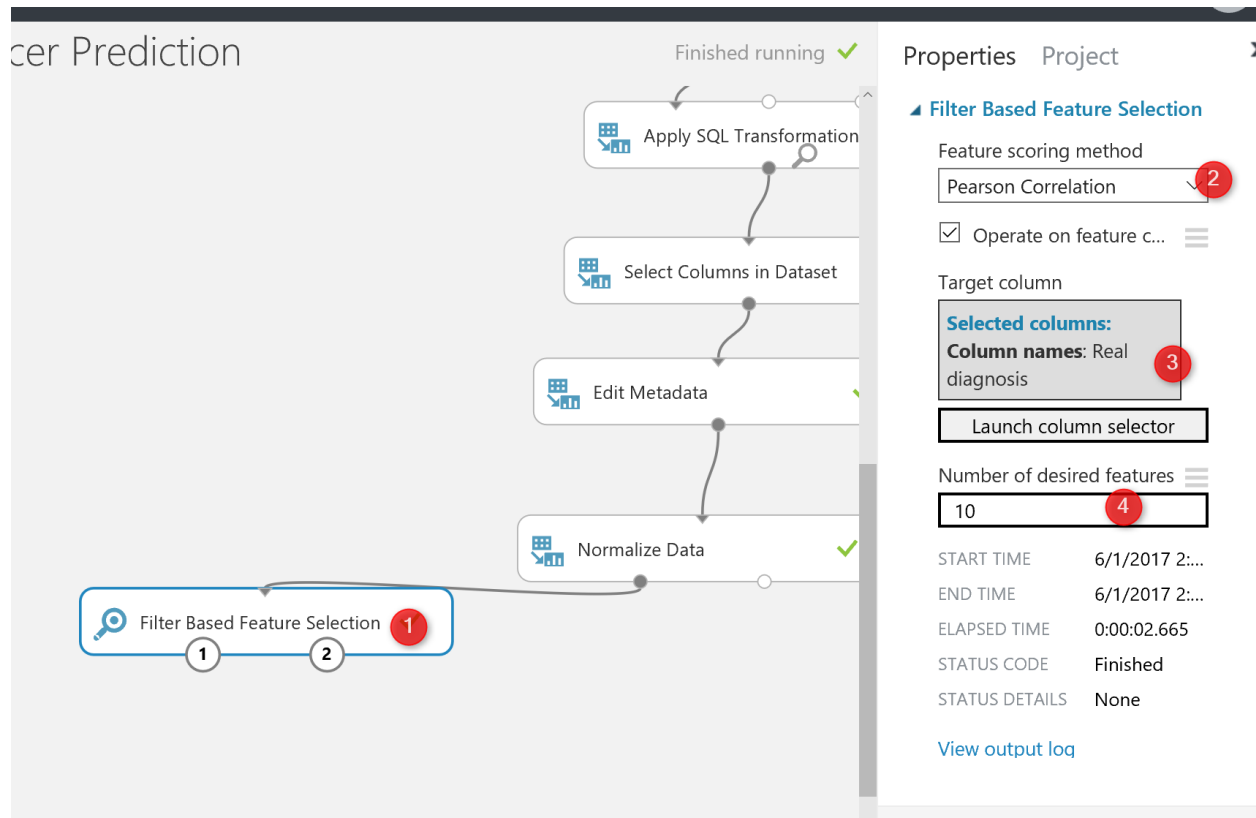
in the below picture, I have shown how I use it to find which attributes have more impact on the diagnosis of the cancer condition.

As you can see in the below picture, I have connected the output of the normalization component to the input node of the “**Filter Based Feature Selection**”. By clicking on this node, in the right side, you will see some options that you have to set them up first.

First of all, you should choose the algorithm for the aim of feature selection. In this experiment, I have choose the “**Pearson Correlation**” analysis (Number 3). However, there are many other approaches that I will talk about them later. Then, in the next textbox (number 4), I have identified the columns that I want to predict, which in our example is “**Diagnosis column**”.



Finally, in the last textbox, I specify the number of features that I am interested in having for prediction among 32 columns, the default value is 1, but I specify it as 10 (see below picture).



The screenshot displays the Azure Machine Learning Studio interface. On the left, a workflow is shown with the following steps: **Filter Based Feature Selection** (marked with a red circle 1), **Normalize Data** (marked with a green checkmark), **Edit Metadata**, **Select Columns in Dataset**, and **Apply SQL Transformation** (marked with a green checkmark). A red circle 2 is placed on the 'Filter Based Feature Selection' node. A red circle 3 is placed on the 'Selected columns' field in the properties pane, and a red circle 4 is placed on the 'Number of desired features' field.

**Properties** Project

**Filter Based Feature Selection**

Feature scoring method  
Pearson Correlation 2

Operate on feature c... ≡

Target column  
**Selected columns:**  
Column names: Real diagnosis 3

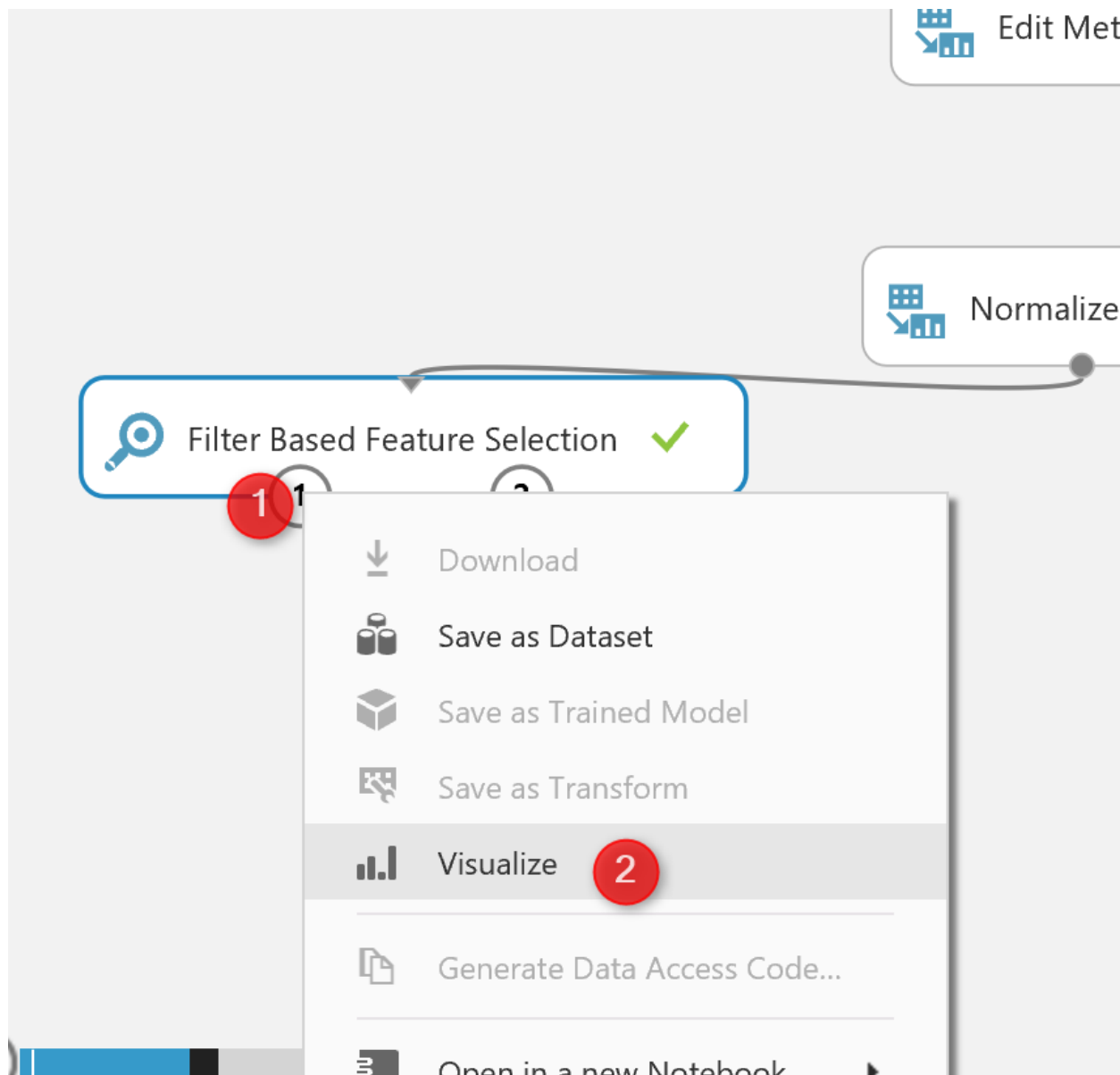
Launch column selector

Number of desired features ≡  
10 4

START TIME 6/1/2017 2:00:00  
END TIME 6/1/2017 2:00:02  
ELAPSED TIME 0:00:02.665  
STATUS CODE Finished  
STATUS DETAILS None

[View output log](#)

Then, I run the experiment to see the result of the feature selection by right click on the left output of the node (see number 1 in the below picture)









The below result will be shown, as you can see, now we have 11 columns instead of the 32 that means these are columns that have more impact on the predicting of a cancer diagnosis.



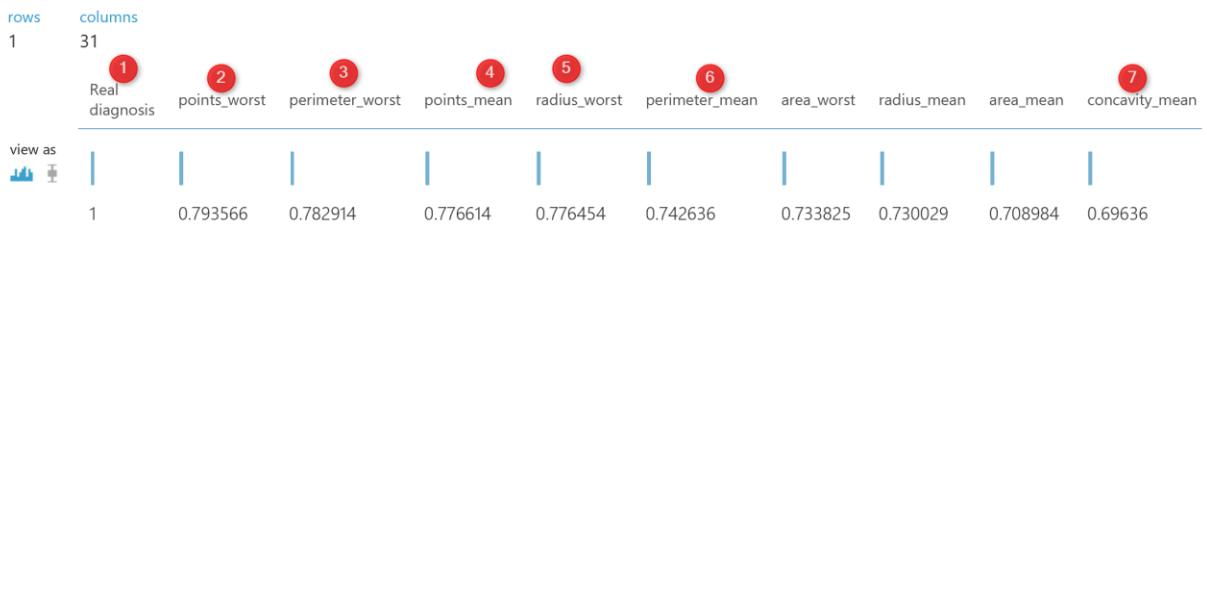
Cancer Prediction > Filter Based Feature Selection > Filtered dataset

rows 569 **1** columns 11 **2**

	Real diagnosis	points_worst	perimeter_worst	points_mean	radius_worst	perimeter_
view as						
	Benign	0.322715	0.182081	0.183897	0.19815	0.242278
	Benign	0.272371	0.138802	0.131312	0.140519	0.176145
	Benign	0.255361	0.147019	0.12326	0.159374	0.187478
	Benign	0.295911	0.130086	0.23837	0.141942	0.201852
	Benign	0.281031	0.269386	0.132058	0.294201	0.372193
	Benign	0.229003	0.179391	0.070974	0.182853	0.210144
	Benign	0.331718	0.158723	0.204026	0.161864	0.212356
	Malignant	0.691753	0.388914	0.456064	0.400925	0.330108
	Benign	0.110069	0.118582	0.059692	0.128424	0.163223
	Benign	0.369416	0.146173	0.166054	0.141942	0.191417
	Malignant	0.726117	0.561731	0.557157	0.566702	0.649644
	Benign	0.282165	0.180238	0.115855	0.201352	0.23827
	Benign	0.405942	0.251457	0.174452	0.244207	0.201540

If you right click on the right side output node of the "feature selection" and visualize the dataset, you will see below data:

Cancer Prediction > Filter Based Feature Selection > Features



This data shows which factor has impacted more on the “real diagnosis” column, for instance, column “Point\_Worst” has 79% impact on the diagnosis, or “Perimeter\_worst” has 78% impact. All of this analysis has been done by correlation analysis to see the impact of each attribute on a predictable column.

We are were done by data cleaning and feature selection. Now we clean our data, we identify which factor has more impact on the “cancer Diagnosis”.

The next step according to the machine learning process is so spat data for training and testing purpose

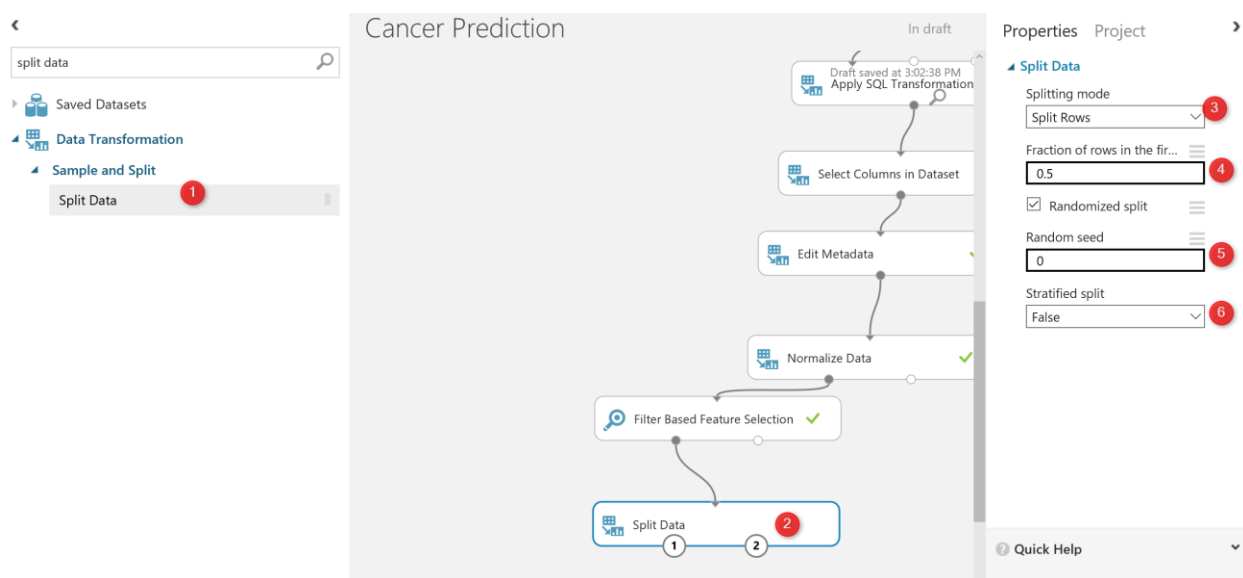
### Split Data

The main aim of machine learning is to learn from past data, so we have to provide a set of data to train the model. The training dataset helps an algorithm to understand the data behavior better, so able to learn from past data and predict future data.

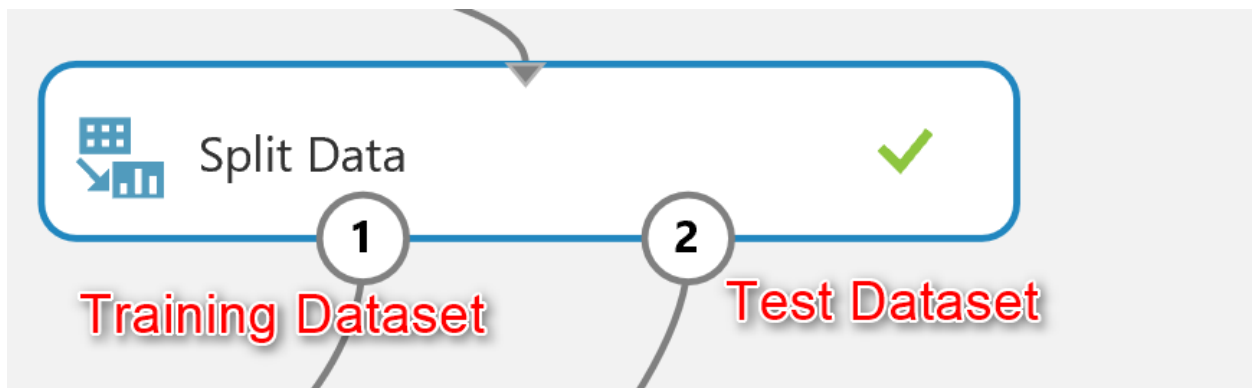
Also, after creating the model, we should test it to see whether they predict well or not, so we have to provide a Test dataset from what we already have to check the results.

There is a component in Azure ML Studio that help us to create a test and train dataset name “**Split Data**”. Split data can be found under the “**Data Transformation**” component. (number 1). Just drag and drop it to the experiment and connect it to the output of “**feature selection**” ( the dataset output that is in left side).

Then on the right side of the experiment, you will see windows that show the parameter list. The first parameter identifies how to split the dataset, which I choose the “Split Rows”, there are other approaches like using a regular expression for dividing the dataset into train and test, which hopefully I will talk about them later. Next in number 4, you see that I specify that 0.5 % of data should go for testing and 0.5% for training. Always this percentage should be above 70% that provides more data for training. In number 5 and six we can set a value for the seed to make the experiment consistency for each run.



Now by running the code, we have two datasets: Training dataset which located on the left side of the “Split Data”. The test data has been located on the right side of the “split Node”

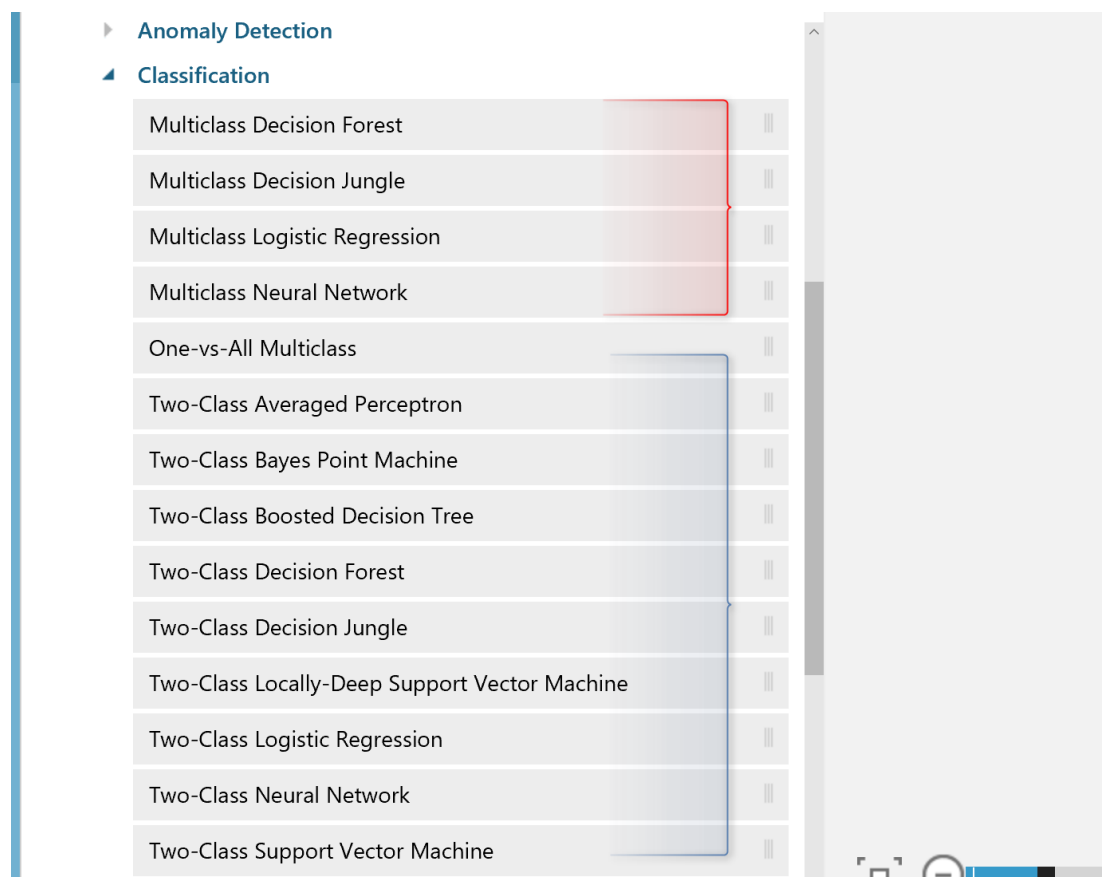


Now we have "Training dataset" and "Testing Dataset".

In the next chapter I will show how to choose algorithms and also how to train, test, and evaluate the model.

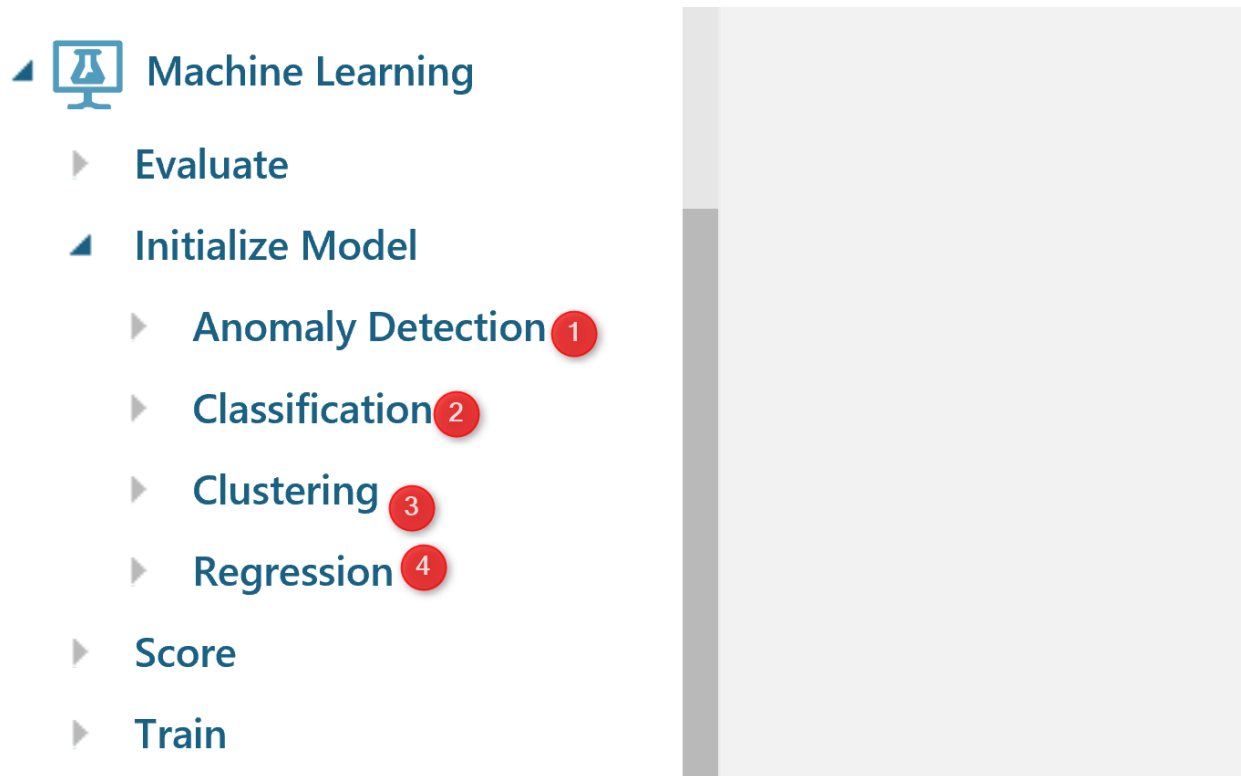
# Chapter 6: A Machine Learning Prediction Scenario (1)

Published Date: June 6, 2017



In previous Chapters (Chapter 4 and Chapter 5), I have explained some of the main components of Azure ML Studio via a prediction scenario. In chapter one the process of data cleaning (using **SQL Transformation, Cleaning Missing Value, Select specific Columns, and Edit Meta Data**) has been explained. And in the second chapter, I have explained how to apply **Data normalization, Feature selection** and how to **split Data** for creating **Training and Testing** Datasets has been explained.

In this chapter, I will continue the scenario of predicting patient diagnosis to show how to select **appropriate algorithms, Train Models, and Test Models (Score)**, In the next chapter I will show how to evaluate a model using **“Evaluate Model” component**, also how to find the best parameters for algorithms using **Tune Model Hyperparameters**, and also check the algorithm evaluation result on different portion of the dataset using **Cross Validate** component.

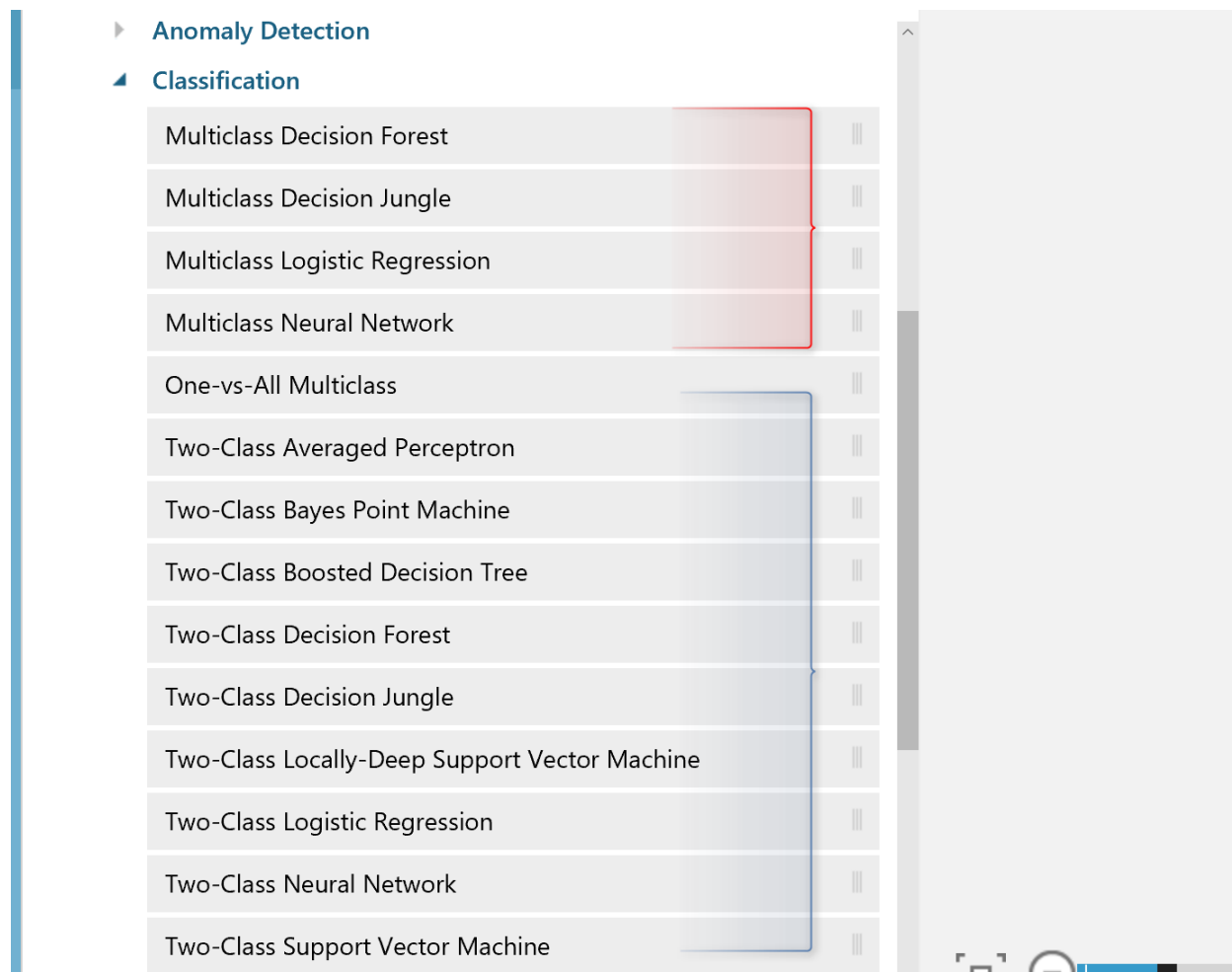


### How to choose Algorithms

Many algorithms address the different types of machine learning problems. In Azure ML Studio, there are four types of algorithms: **Anomaly Detection, Classification, Clustering, and Regression.**

**Anomaly Detection:** these types of algorithms help find cases that not follow the normal pattern of the data such as finding the fraud in credit cards (above picture Number 1).

**Classifications:** Classification can be used for predicting a group. For instance, we want to predict whether a customer will **stay** with us **or not**, or a customer will get a **Bronze, Silver, or Gold** membership. In Azure ML Studio we have two sets of algorithms for classification as: **Two-Class and Multi-class algorithms** (See below Picture)



### Clustering

Clustering algorithms like k-mean clustering (I have a chapter on it) more used for finding the natural pattern in data without making a prediction.

### Regression

Regression algorithms are used for predicting a value. For instance, predicting the sales amount in a company.

In this scenario we are going to predict whether a patient will be **Benign or Malignant**. So, I have to use two-class algorithms.

So, I am going to use **two-class classification algorithms**.

To get a better result, it always recommends to :

**Try different algorithms** on a dataset to see which one better able to predict.

**Try different Dataset** it is a good idea to divide a current dataset to multiple one, and then check the algorithms evaluation in different Chapters of a dataset. in this scenario I am going to use a component name "**Cross Validate Model**"

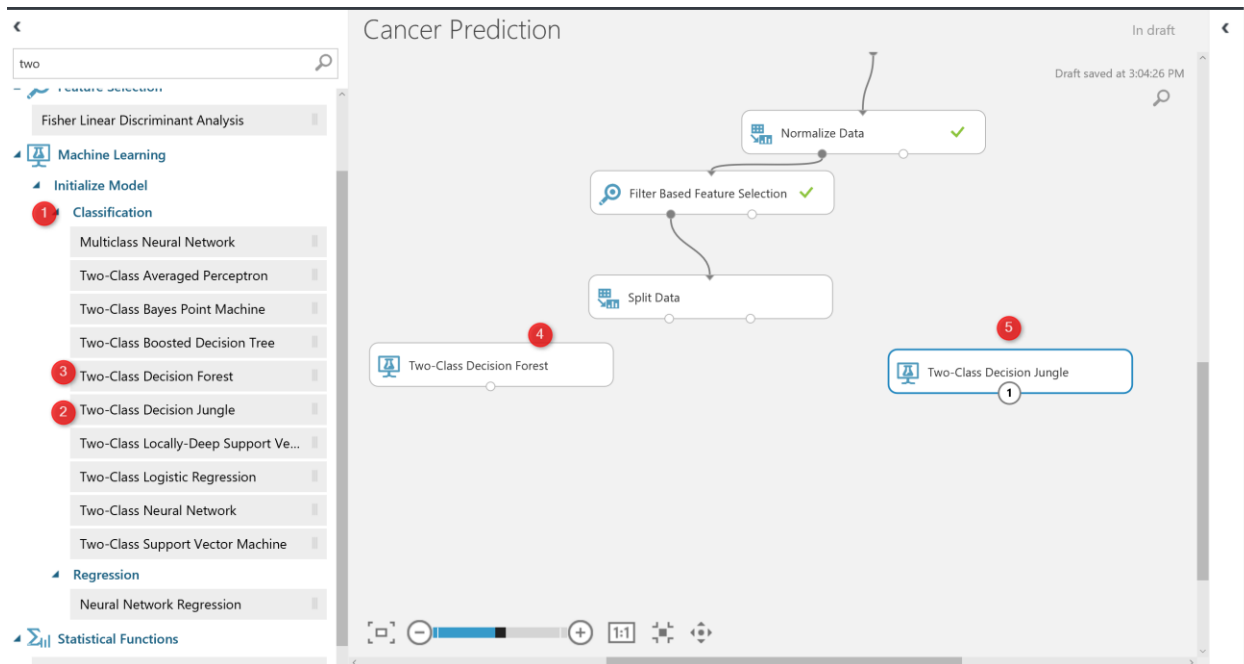
**Try different algorithms Parameters.** Each algorithm has a specific parameter's values, It always recommended to try the different parameter's value. There is a component in Azure ML Studio called "**Tune Model Hyperparameters**", that run the algorithms against different parameter's value.

First I am going to try two algorithms on a single dataset to see which one better predict the patient's diagnosis.

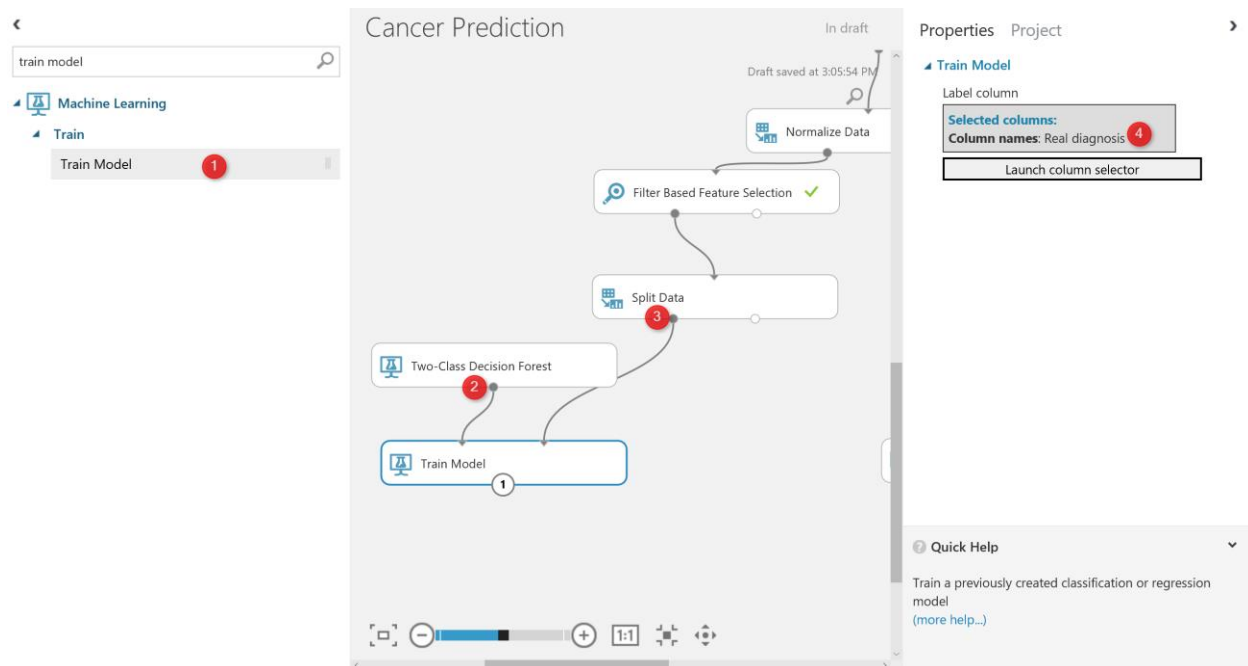
I select the "**Two-Class Decision Forest**" and "**Two-class Decision Jungle**" to see which one better predict.

I search for them in the left side menu, and drag and drop them into the experiment area.

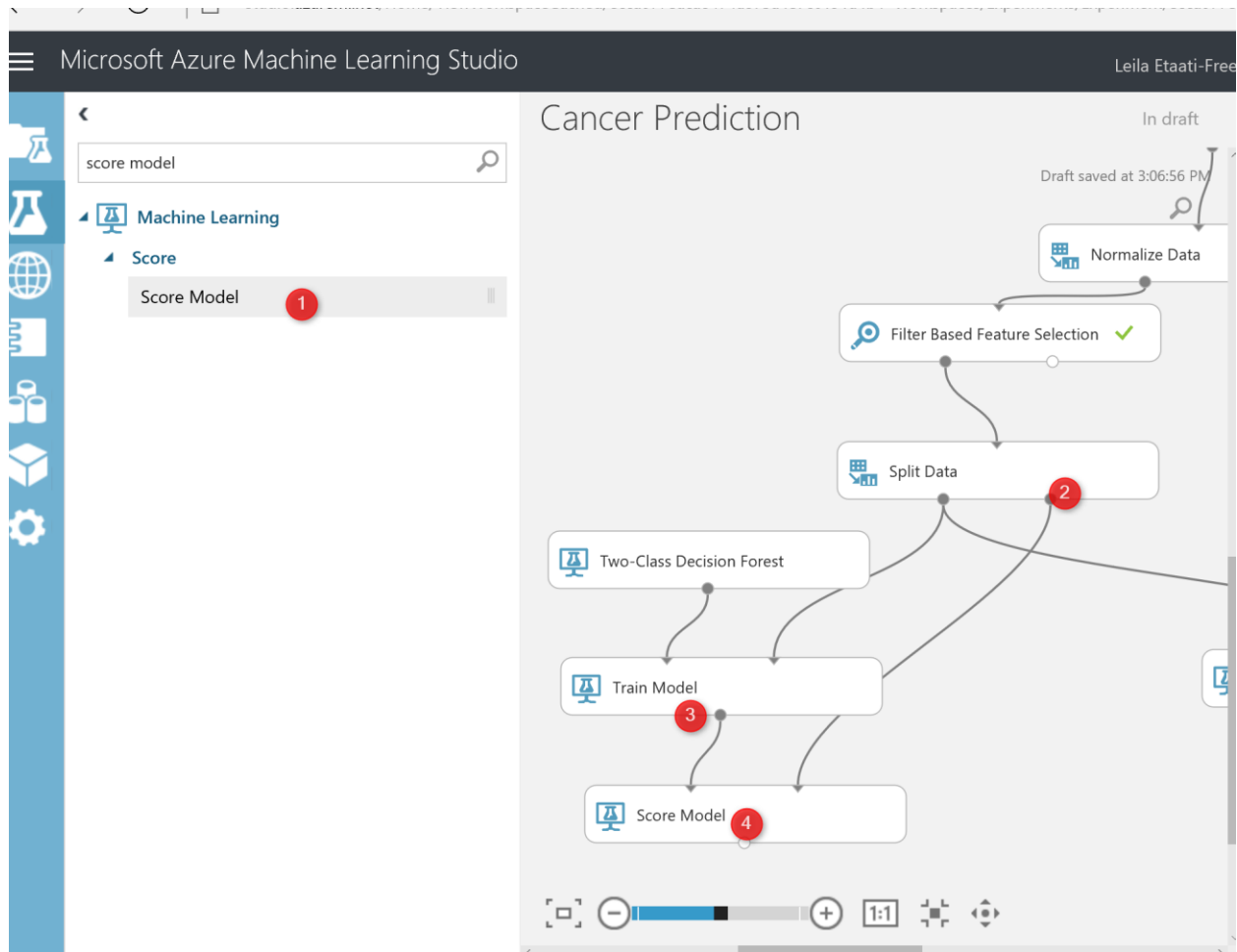




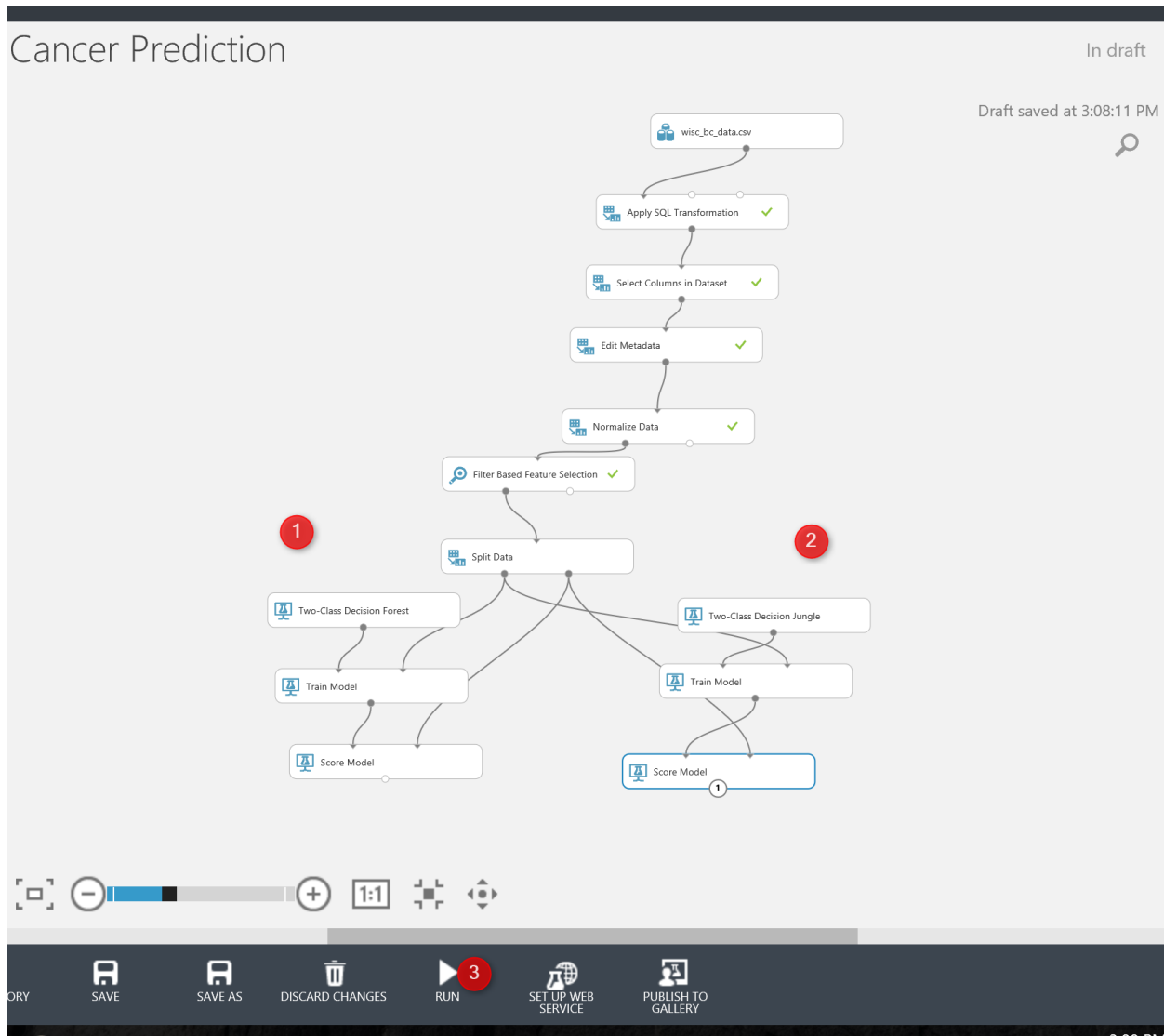
to work with these models we need to train them first, there is a component name **“Train Model”** that helps us to train a model. The Train Model component gets two inputs: one from the algorithm and the other from the dataset (in our example from a split output node at the left side) see below picture. Moreover, click on the train model components, and on the right side of the experiment in properties (number 4), we have to specify the columns that we want to predict. In this example is **“Real Diagnosis”**.



Finally, we have to test the model. To test a model in Azure ML Studio, there is a component name **“Score Model”**. Search for it and drag and drop it into the experiment area. The scoring model gets first input from train model component, that is a machine learning algorithm, and the right side input from the split component the output for the testing dataset.



I just created the same process for **“two-class decision jungle and forest”** (see below picture). Now I am going to run the experiment, to see the result.



after running the experiment, click on the output node of one of the scoring model and visualize the dataset. In this dataset, you will see that we have 284 rows (that is the testing dataset), also you will see that we have 13 columns instead of 11. the two other columns are for the result of predictions as: **Score Label** and **Score Probabilities**.

The Score Label: show the prediction of the diagnosis for patients. The score Probabilities show the probability of predictions. For instance, for the first row the prediction of the model was patient with these laboratory result will be **Benign** with 50% **probability**

Cancer Prediction > Score Model > Scored dataset

rows 284 columns 13

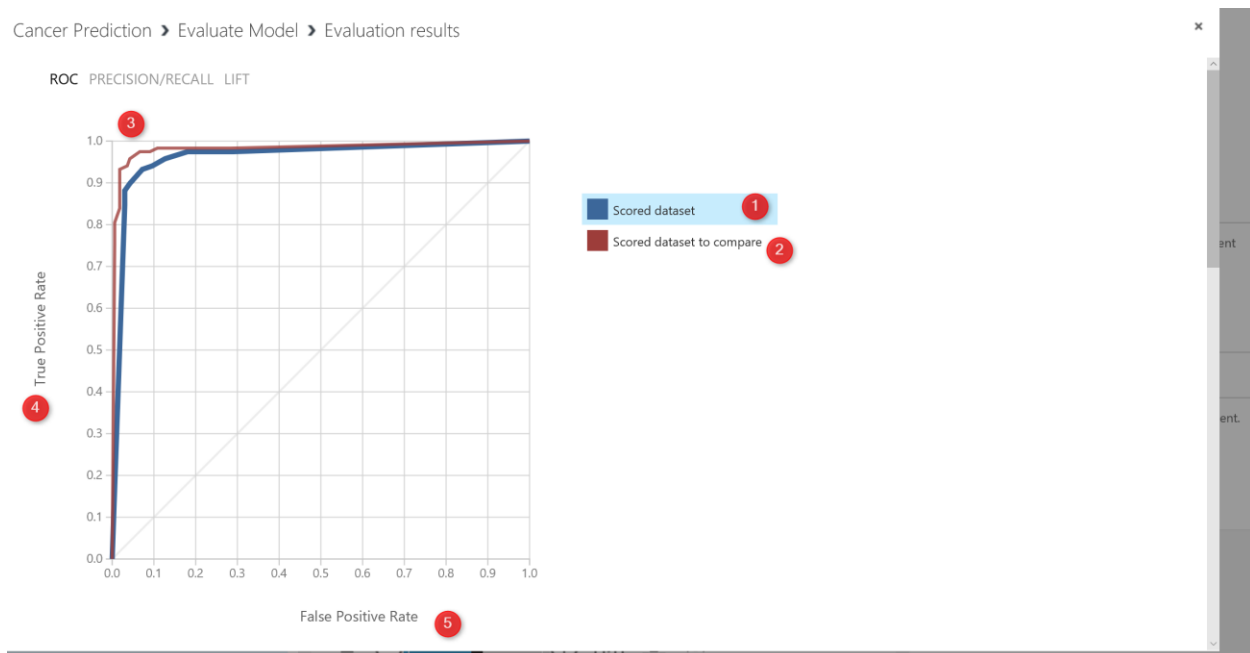
er\_worst points\_mean radius\_worst perimeter\_mean area\_worst radius\_mean area\_mean concavity\_mean concavity\_worst Scored Labels Scored Probabilities

	er_worst	points_mean	radius_worst	perimeter_mean	area_worst	radius_mean	area_mean	concavity_mean	concavity_worst	Scored Labels	Scored Probabilities
36	0.151988	0.303095	0.335982	0.158106	0.341663	0.201442	0.118627	0.200639	Benign	0.5	
41	0.328131	0.278904	0.360998	0.143064	0.351602	0.21527	0.319119	0.177316	Malignant	0.625	
77	0.916998	0.78513	0.882524	0.584153	0.863221	0.735737	0.782334	0.517252	Malignant	1	
52	0.22505	0.347919	0.418147	0.187451	0.43348	0.278473	0.128866	0.184505	Malignant	1	
89	0.614811	0.82106	0.758137	0.678038	0.768091	0.647508	0.456888	0.464856	Malignant	1	
93	0.257555	0.316258	0.23592	0.172901	0.229968	0.126023	0.28538	0.555591	Malignant	1	
39	0.031978	0.157595	0.18658	0.070217	0.200625	0.103203	0.011638	0.018514	Benign	0	
15	0.299105	0.42974	0.458227	0.256538	0.461877	0.307953	0.315839	0.432029	Malignant	1	
	0.839463	1	0.988943	1	0.967343	1	0.851687	0.545767	Malignant	1	
6	0.444384	0.62042	0.628913	0.432757	0.646457	0.505408	0.357779	0.332188	Malignant	1	
73	0.074751	0.088225	0.112777	0.03576	0.117564	0.053404	0.05089	0.075176	Benign	0	
9	0.162227	0.253291	0.294036	0.1279	0.302854	0.175483	0.06605	0.055935	Benign	0	
83	1	0.643543	0.845899	0.46397	0.817313	0.686108	1	0.463498	Malignant	1	

In the next chapter, I will show How to **evaluate** the result of the prediction, how to try different algorithm **parameters** using hyper tune parameters, and also how to check different dataset using **Cross-Validation**.

# Chapter 7: A Machine Learning Prediction Scenario (2)

Published Date: June 8, 2017

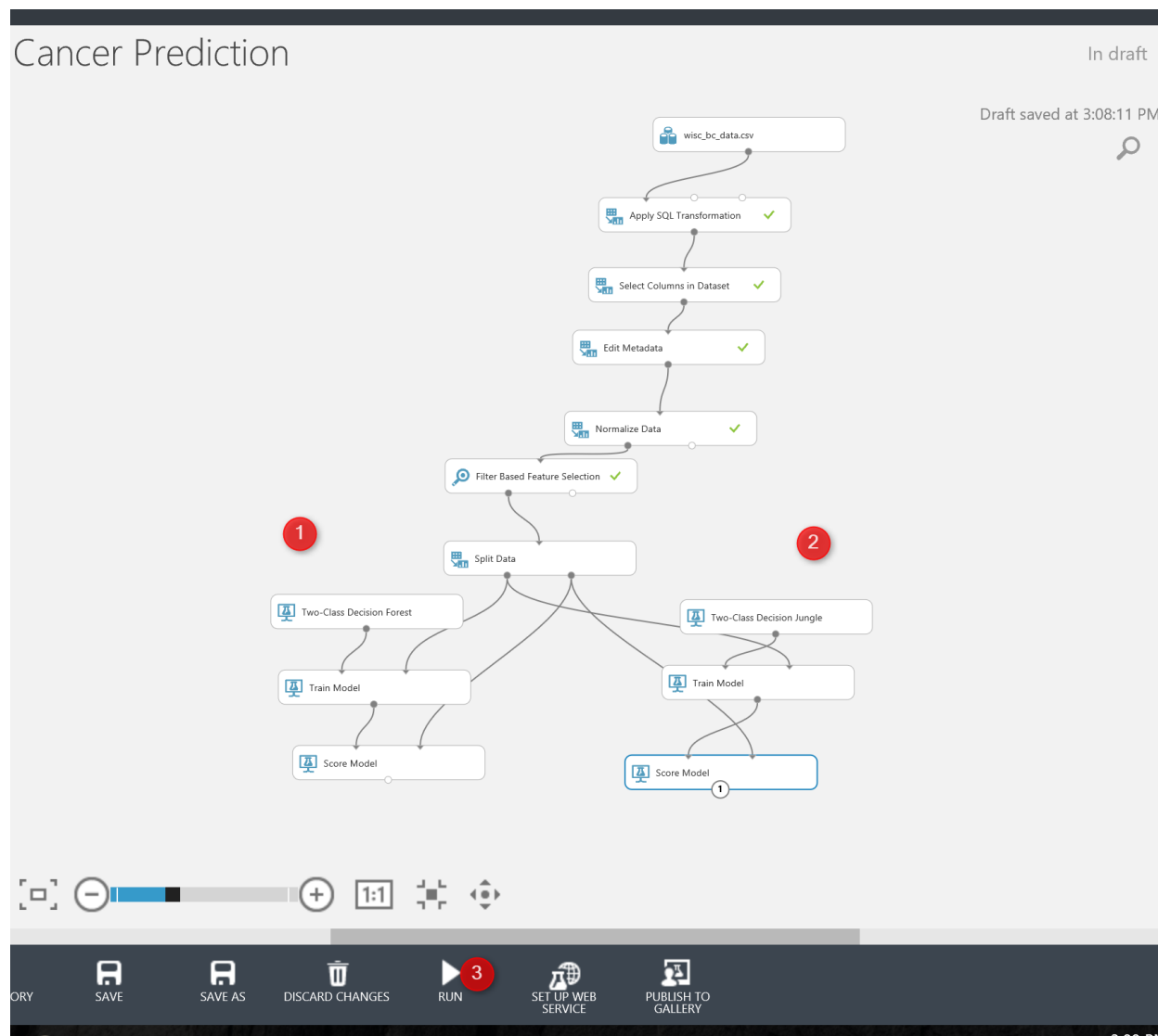


In the previous Chapters from Chapter 1 to 6, I have explained how to do machine learning process.

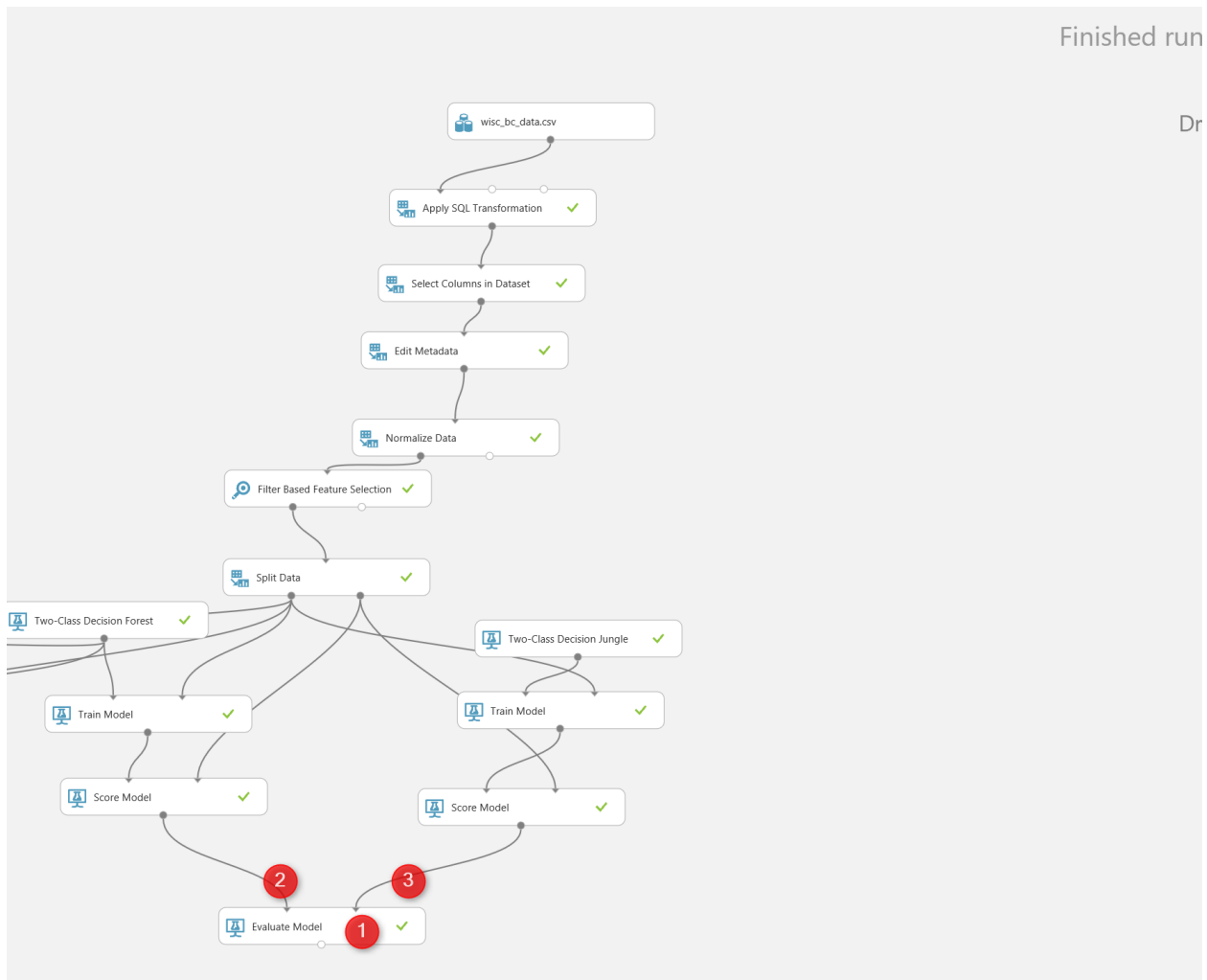
The data cleaning such as **SQL transformation**, **select specific columns**, **remove missing values**, **Edit metadata**, and **normalize data**. Also, I have explained how to find relevant attributes using **Feature Selection Feature** to identify which feature is more important than the other. Then I showed how to split data using **Split Data** component to train and test the model. Then, I show how to train and test the model.

In this chapter, I will show how to evaluate the results and also how to interpret the evaluation measures.

So in the previous Chapter we came up with the below process. we create a model (see below picture)



Now we need to evaluate the model. There is a component name **Evaluate Model**. Just drag and drop it to the experiment area (see below picture). Evaluate model has two input node. In this experiment, we have two models, so we have two score model (as you can see in the below picture). Then connect the output node of Score Model (Testing) to the input of **Evaluate Model** (see number 2 and 3).



Now let's check the Evaluate model components.

Right click on the output node of the **Evaluate Model** to see the result of the evaluation via **Visualize icon**. You will see the below image right after. We have a chart below — the legend shows to color as blue and red. The blue one is related to the left side algorithm (in our experiment) and the red one related to the right side algorithm) used to check the performance of a two-class algorithm.

In this chart the first chart is "ROC" is a **receiver operating characteristic curve**. The plot is used to check the performance of the binary classification. We have two value in x and Y axis — **true Positive and False Positive Rate**.



But, what is **True Positive and False Positive**?

In a Two-Classification Problem, the prediction and real world scenario may have below situation. The evaluation will look at the test dataset to compare the

**True Positive:** if values in the real world are 1 (for instance in our example, Benign) and our algorithm correctly predicts patients will be Benign so we call it true positive (the higher number better) 😊

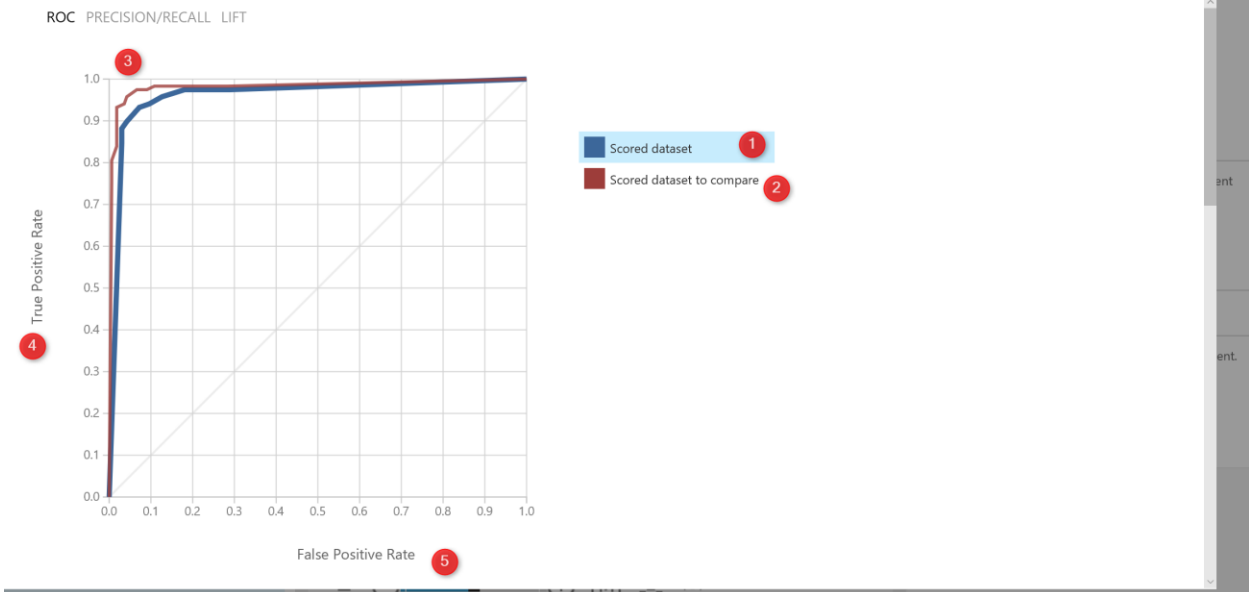
**False Negative:** In the real world scenarios, the patient becomes Benign but in our algorithm predicts they will get Malignant. Which means in real case it is true but in Prediction it is Negative. 😞

**False Positive:** In a real-world scenario, the patient becomes Malignant but in our model, we predict they are Benign which is incorrect so we call it False Positive. 😞

**True Negative:** In a real-world scenario, the patient becomes Malignant and our machine learning prediction is that they will become Malignant 😊

So in the **ROC** chart as below we are going to see the relation between True Positive and False Positive. As you can see below picture, as the line close to the Y axis, it means that it has better performance. That is below picture both charts have good performance.

Cancer Prediction > Evaluate Model > Evaluation results



If you scroll down the evaluate model page, you will see the below picture . as you can see we have some values that help us to evaluate the model's performance.

First we have the True Positive, False Negative, true negative and false positive value on the right side of the page (below picture number 1).

Also we have some other measure for evaluating the model performance:

**Accuracy:** The accuracy is about the  $(\sum TP + \sum TN) / \text{Total population}$

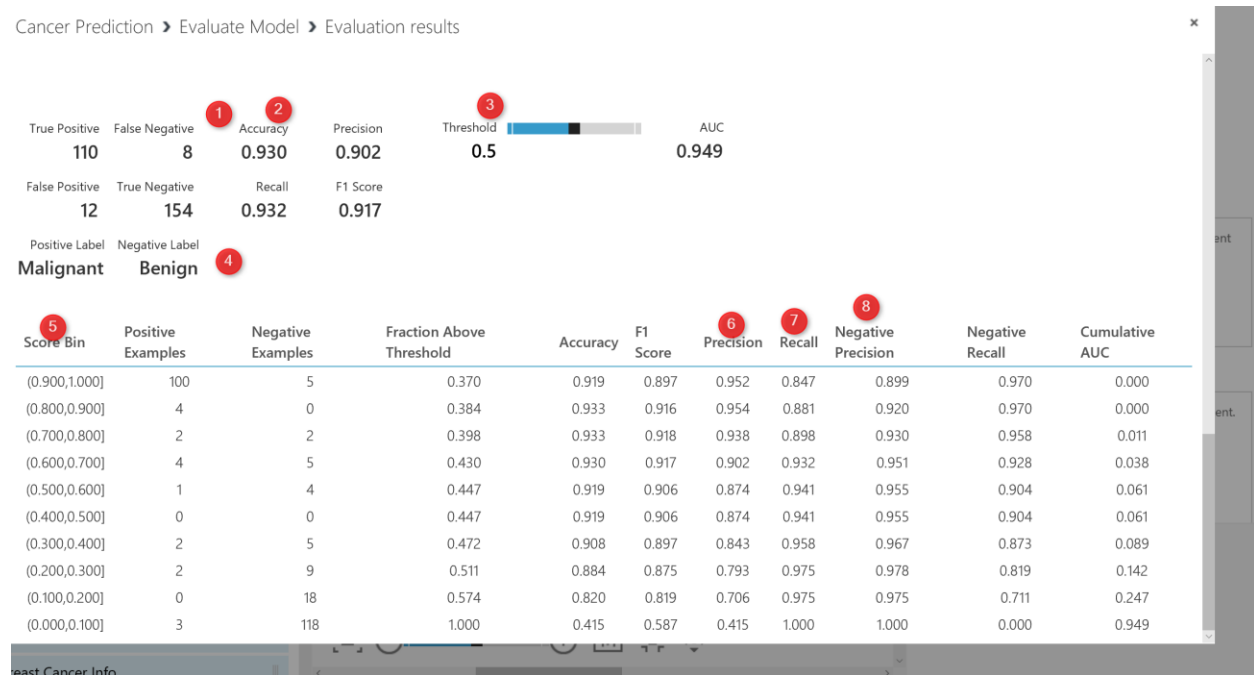
So in the below picture in number 2, you will see it is 93%. As much as accuracy higher better.

**Recall:** The recall is the  $\sum TP / (\sum TP + \sum FN)$ . That measure how the model is good in predicting the true positive cases comparing the total case that happen true. In our example it is 93%.

**Precision:** The precision is the  $\sum TP / (\sum TP + \sum FP)$  . in our example is 90%

**F1 Score:** The F1 Score is the  $\sum 2TP / (\sum 2TP + \sum FP + \sum FN)$  which is 91%.

The bar chart for **Threshold and AUC** (the area under the ROC curve) shown in below picture. It shows 94% which is great. The Threshold of 50 %. So it always good to have an AUC more than the threshold, which means truer positive than False Positive.



Overall, all two algorithms perform well.

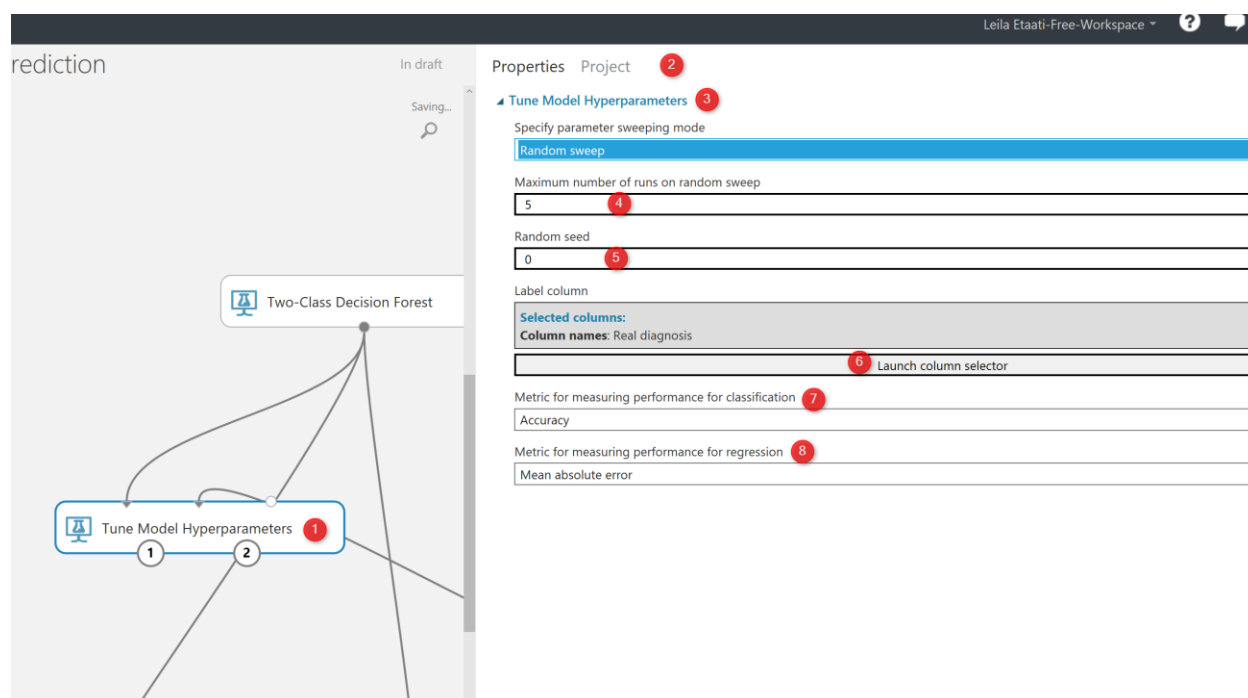
As I have mentioned it, to enhance the performance algorithms, there are three main approaches:

1. it always recommend to run **multiple algorithms** to see which algorithms best fir the dataset and able to predict the data better.
2. Do **Cross Validation**, to check the result of Machine Learning on each dataset sample
3. Check the different parameters for algorithms. Each algorithm has its parameters list, it is important to find the best parameters that cause better performance. This can be done in Azure ML Studio via **Hyper Tuning Parameters**.

In the next chapter, I will show how to enhance the performance of model via **Cross Validation and Hyper Tune Parameter** Component.

# Chapter 8: Tune Parameters: Machine Learning Prediction

Published Date: June 19, 2017

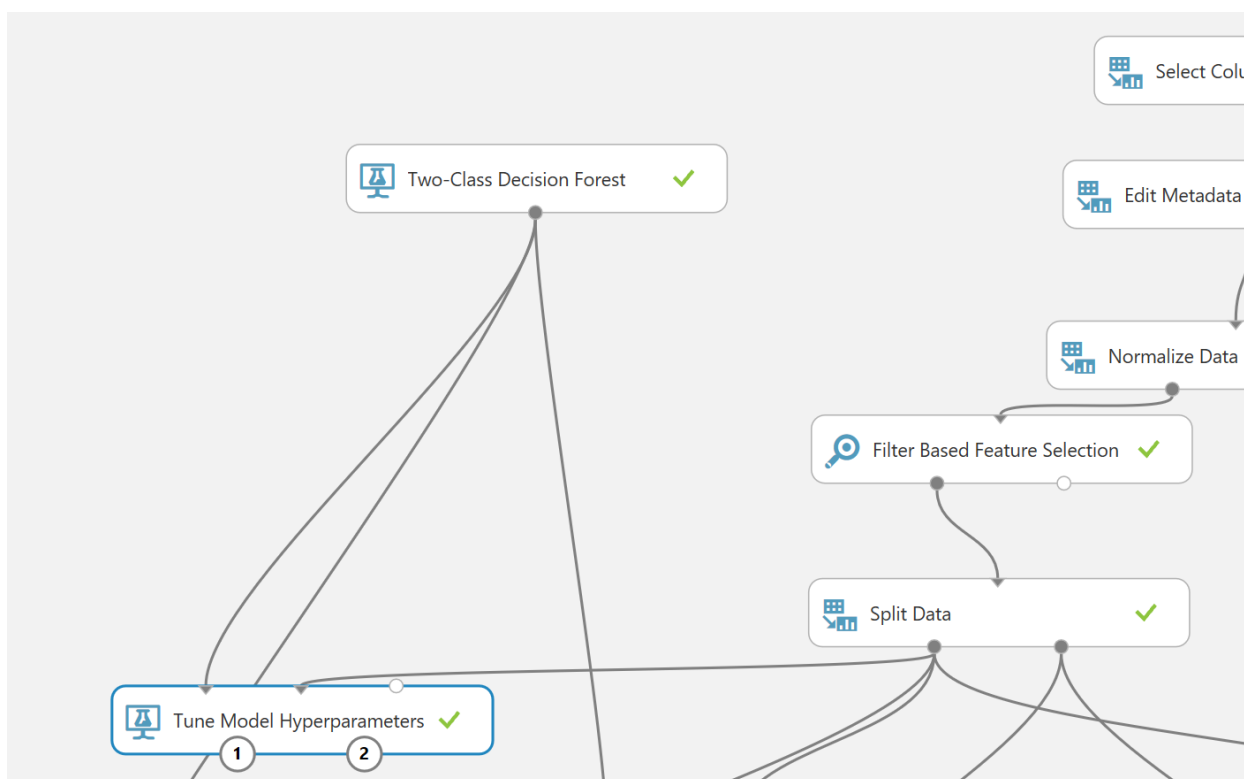


In the previous Chapters from Chapter 1 to [7](#), I have explained how to do machine learning with Azure ML Studio. I have explained some of the main components in Azure ML Studio that helps us to do data wrangling, train the model, feature selection and evaluating the result.

The data cleaning such as **SQL transformation**, **select specific columns**, **remove missing values**, **Edit metadata**, and **normalize data**. Also, I have explained how to find relevant attributes using **Feature Selection Feature** to identify which feature is more important than the

other. Then, I showed how to split data using **Split Data** component to train and test the model. Then, I show how to train and test the model. Moreover, in the last chapter I have explained the evaluation process **Evaluate Model**.

In this chapter, I am going to show another way of enhancing the model as **“Try Different parameters”**. Each algorithm has its parameters. Choosing the right Parameters for each dataset and algorithms can improve accuracy. In Azure ML Studio, there is a component name **“Tune Hyper Parameters”**. This component will help us to improve the accuracy better.

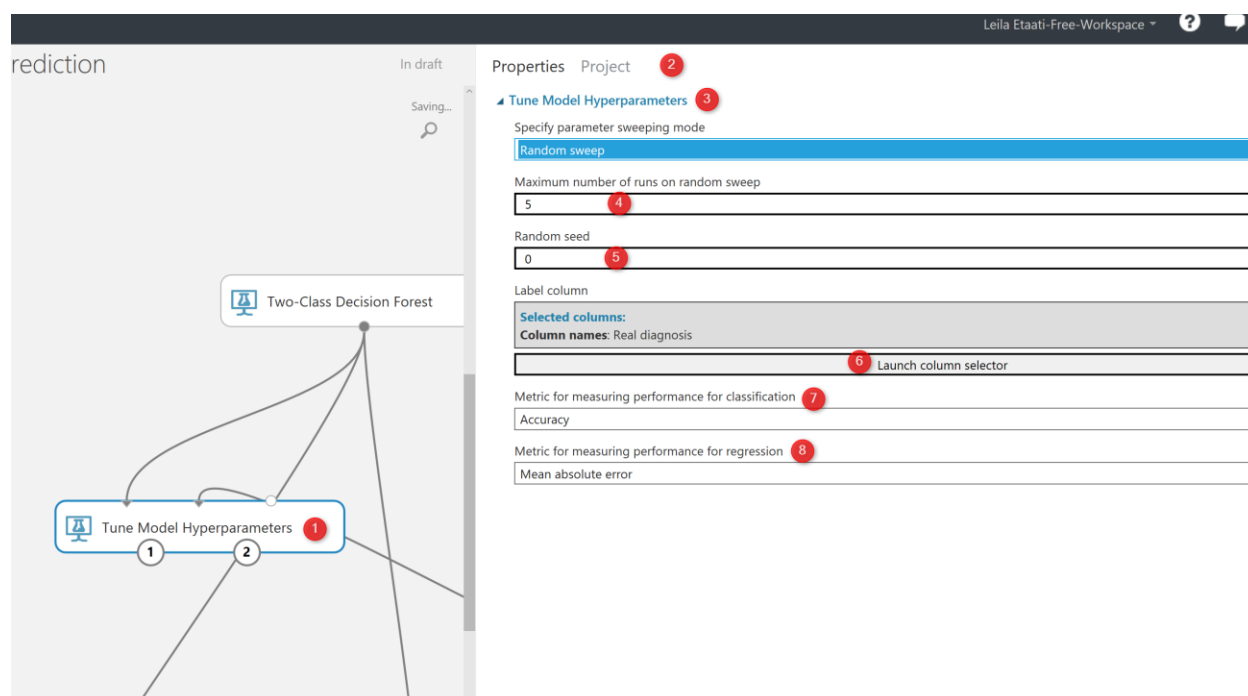


**“Tune Model Hyperparameters”** get two inputs: one for data and for the aim of the training model, which comes from the split data component, another one from the algorithm. This component can be the replacement of **the Train Model**.

If you click on the component, on the right side of the experiment area, you will see the properties panel. As you can see in the below picture, by clicking on the **“Tune Model Hyperparameters”**. We have a couple of the parameters that we should set them up. The first parameter is about the

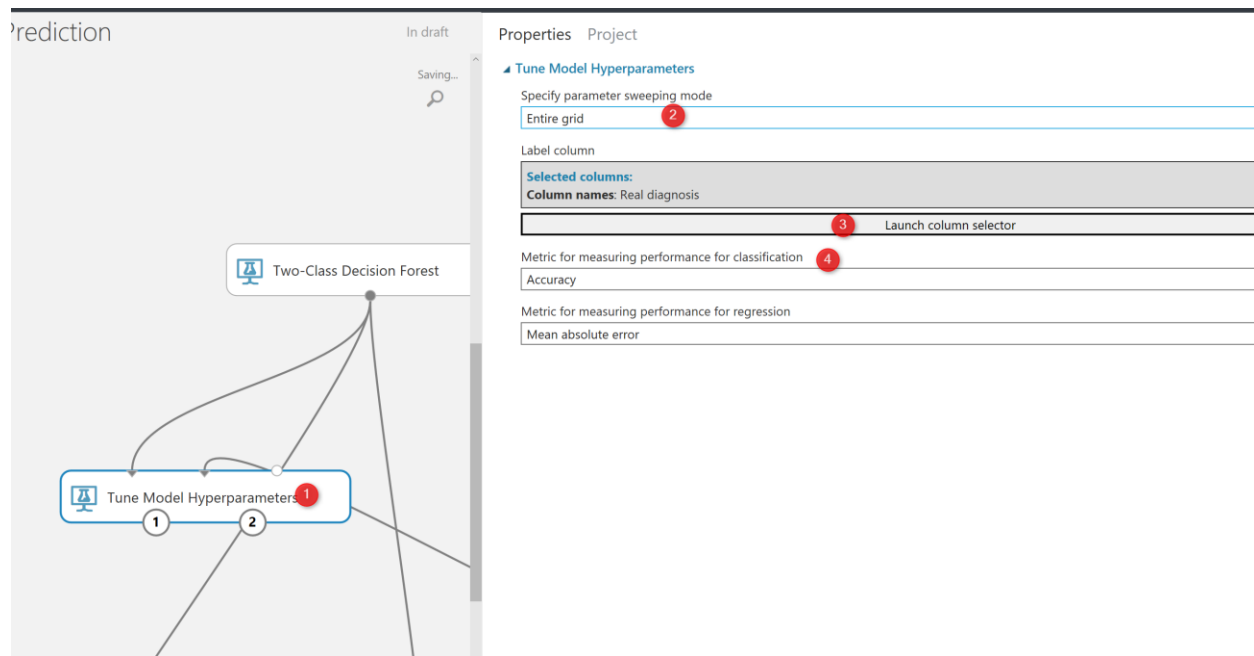
method that we want to try different parameters value. In the below picture, first I have chosen the **“Random Sweep”**. It performs a set number of training iterations by randomly choosing a parameter’s value. So, this component will try different parameter’s value randomly and train the model based on them(Number 3). The second parameters to set up is (number 4) how to specify the number of times we have to run the code. This helps us to identify with which parameters what accuracy we will have.

Then in number 6, we have to specify the item we are going to train the model and identify which column we are going to predict. For this example we choose **“Real Diagnosis”**. In number 7 and 8, we have to specify the metric for measuring the model performance. If the problem is classification, then we need to go for accuracy or recall measures. If the model is about predicting a value and we using the regression models, then we should select one of the items from number 8 in the picture.



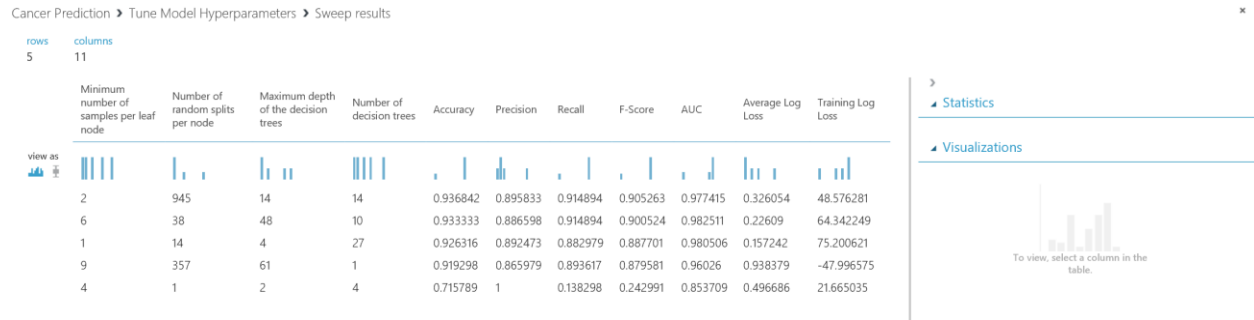
The other option is to test all the parameters against each other that means trying different parameters combinations and identifying the best of them. We call this approach as trying **“Entire Grid”**. as you can see in the below picture in number 1, we select **“Entire Grid”** as the approach to

find the best parameters. As you see here we are not to specify the number of times we run the training model.

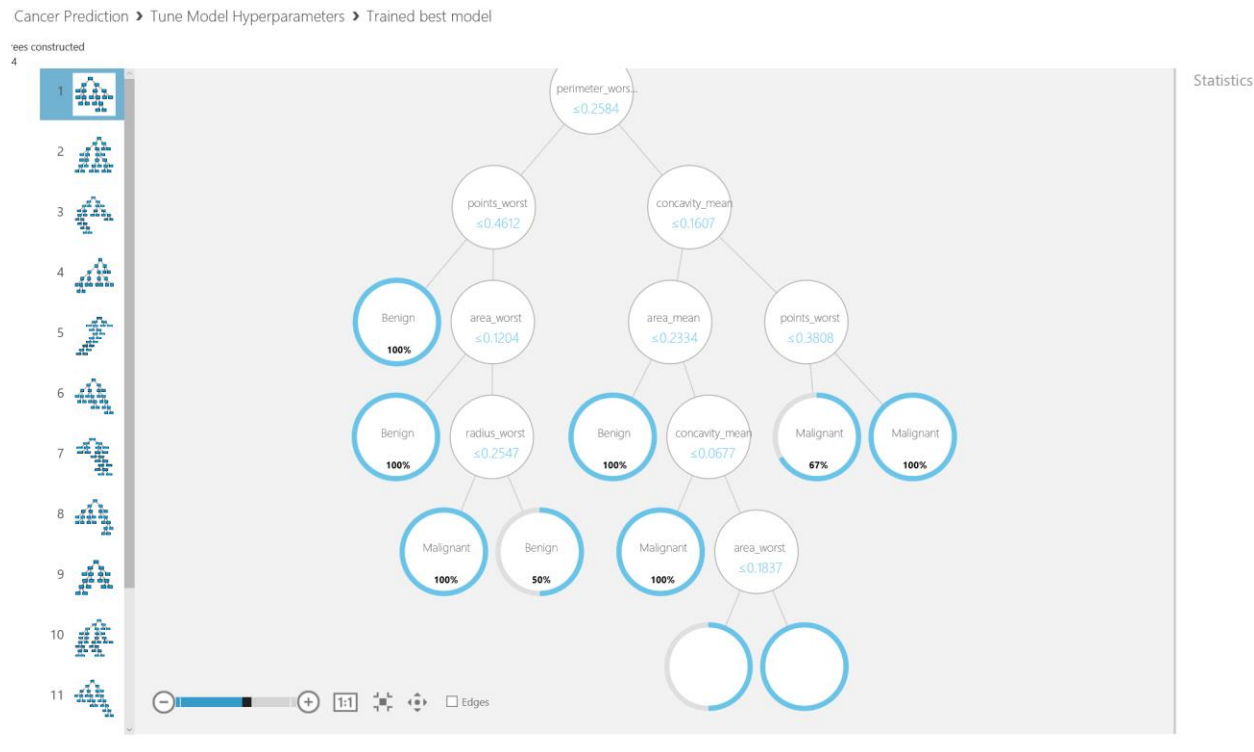


Now, we set up the code, we are going just to run the experiment. After running the experiment you will see we have two output for the "Tune model Hyperparameters". The first output at the left side has the dataset that shows the parameters value and relative accuracy to them (See below picture).

As you can see in the below picture, the first 4 first column is parameters for decision forest algorithm that we have. Such as the number of sample per leaf, depth of the decision tree, number of the decision tree. As you can see from the below picture, we have six other columns that shows the accuracy, Precision, Recall, and so forth value that we got based on the selected parameters. So it helps us to assign better parameters for our model.



The other output of the "Tune Model Hyperparameter" is to visualize of the decision tree (see below picture). So you see the different models that have been trained.



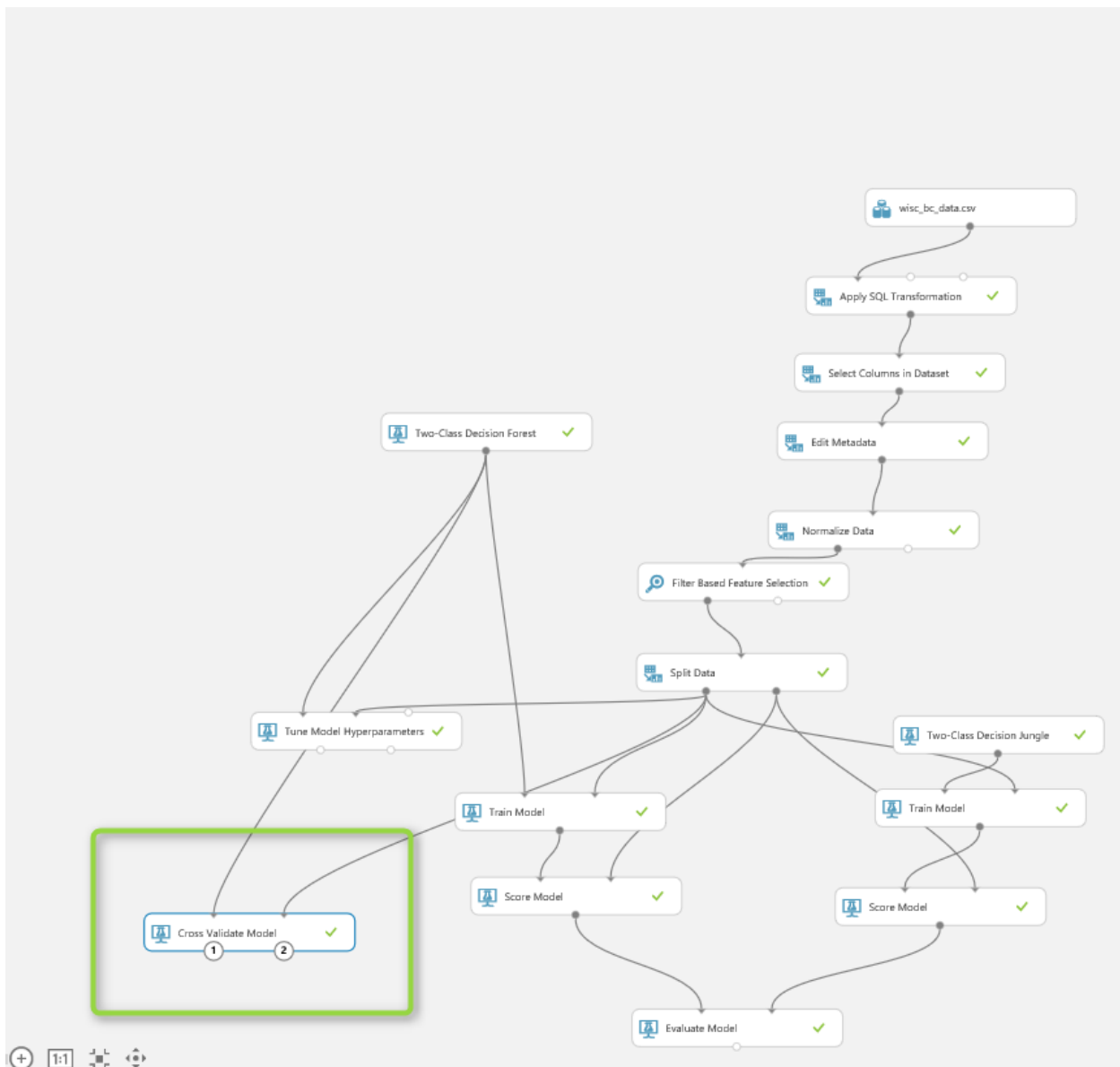
In the next chapter, I will talk about the "Cross Validation" component that is another way of enhancing the accuracy by trying different datasets.

<https://msdn.microsoft.com/library/azure/038d91b6-c2f2-42a1-9215-1f2c20ed1b40>



# Chapter 9: Cross Validation: Machine Learning Prediction

Published Date: June 20, 2017

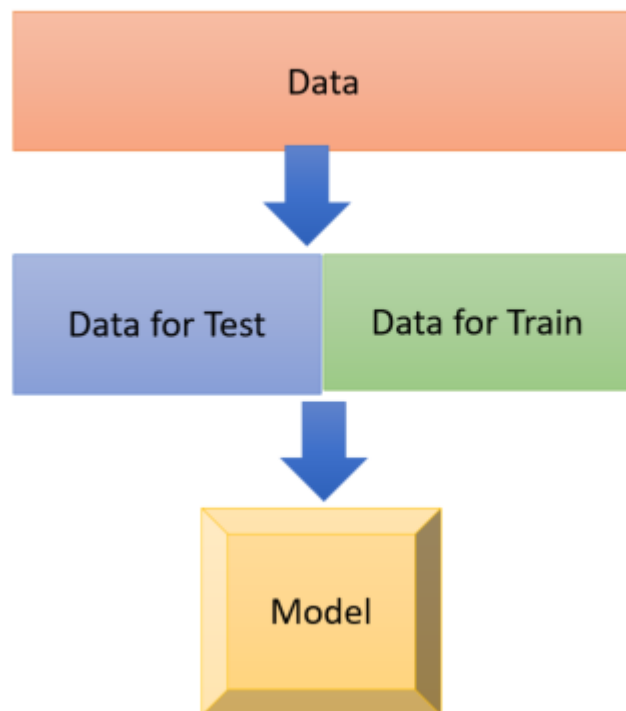


In the previous Chapters (from Chapter 1 to Chapter7), I have explained the whole process of doing machine learning inside the Azure ML Studio, from import data, data cleaning, feature selection,

training models, testing models, and evaluating. In the last [chapter](#), I have explained one of the main ways of improving the algorithms performance name as “**Tune the algorithm’s parameters**” using “**Tune Model Hyperparameters**”. In this chapter am going to show another way for enhancing the performance using “**Cross Validation**”.

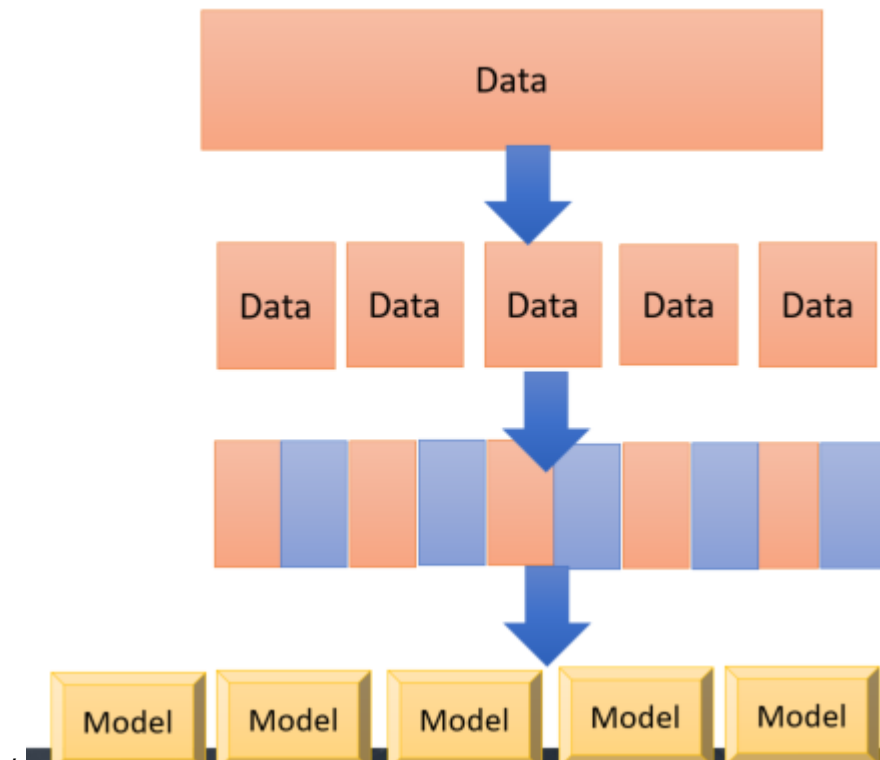
## So what is Cross Validation?

Cross-Validation is a model validation technique for assessing how the results of the statistical analysis will generalize to an independent data set. So we are applying the algorithms in a different portion of the dataset to check the performance. So the process is 1. get the untrained dataset and model 2. partition data into some folds and sub-datasets 3. applied the algorithms on this dataset separately it helps us to see how algorithm perform for each dataset, also it helps us to identify the quality of the data set and understand whether the model is susceptible to variations in the data. So initially, we follow the below process. Just split the dataset into two separate datasets as below.

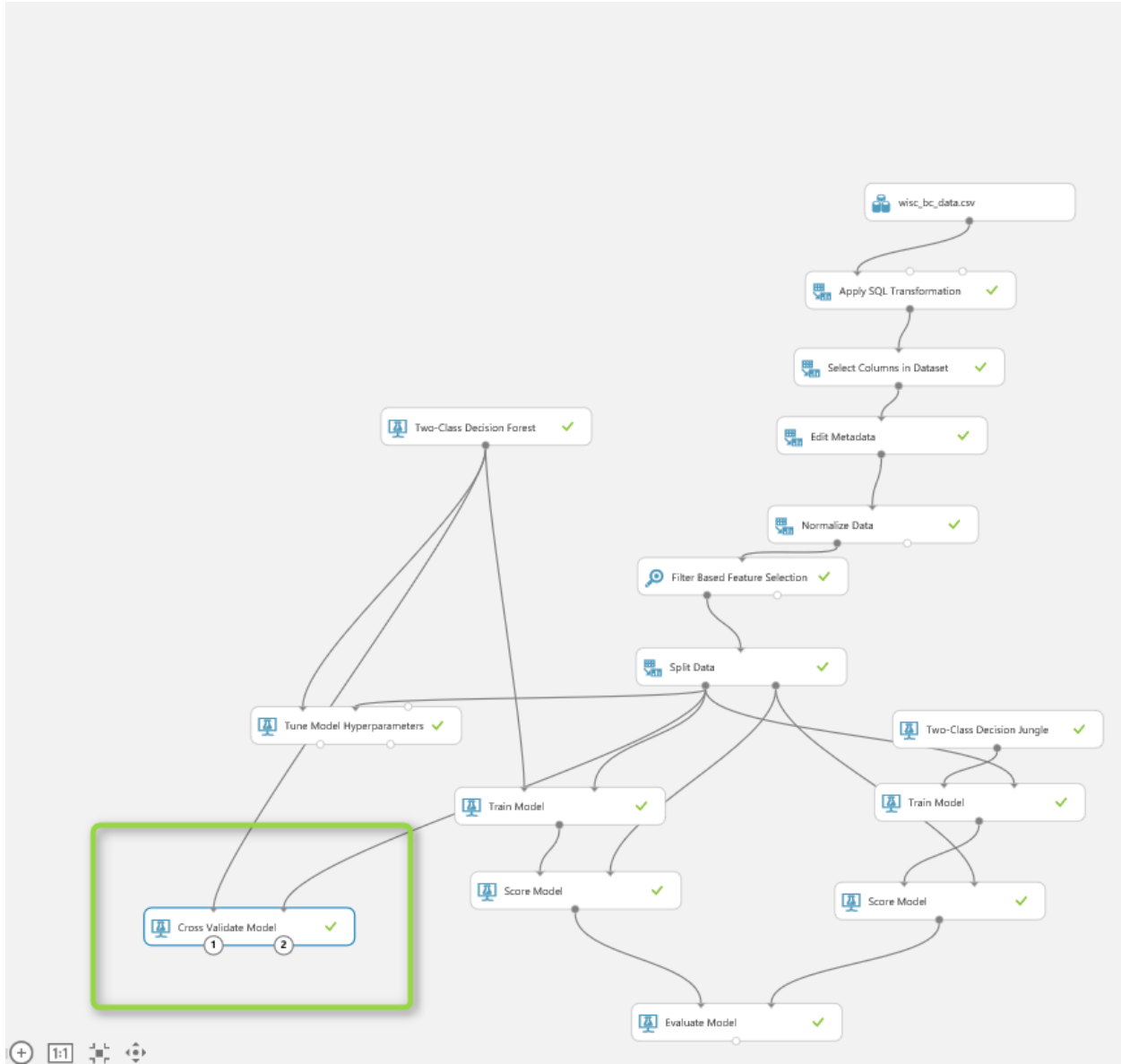


One for Training and one for testing.

Now we are going to divide our datasets into multiple folds as below picture. As you can see in the below picture, I have five different datasets, which I applied a model to each of them to see what is the performance on each dataset

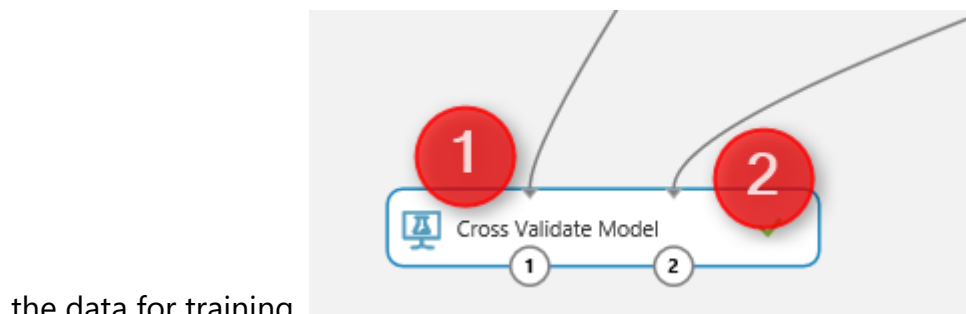


In Azure ML Studio, we able to do the Cross-Validation with the component with the same name as "**Cross Validation Model**". Cross-Validation model works both for training and scoring (testing) the model, so there is no need to add these components (See the below picture)



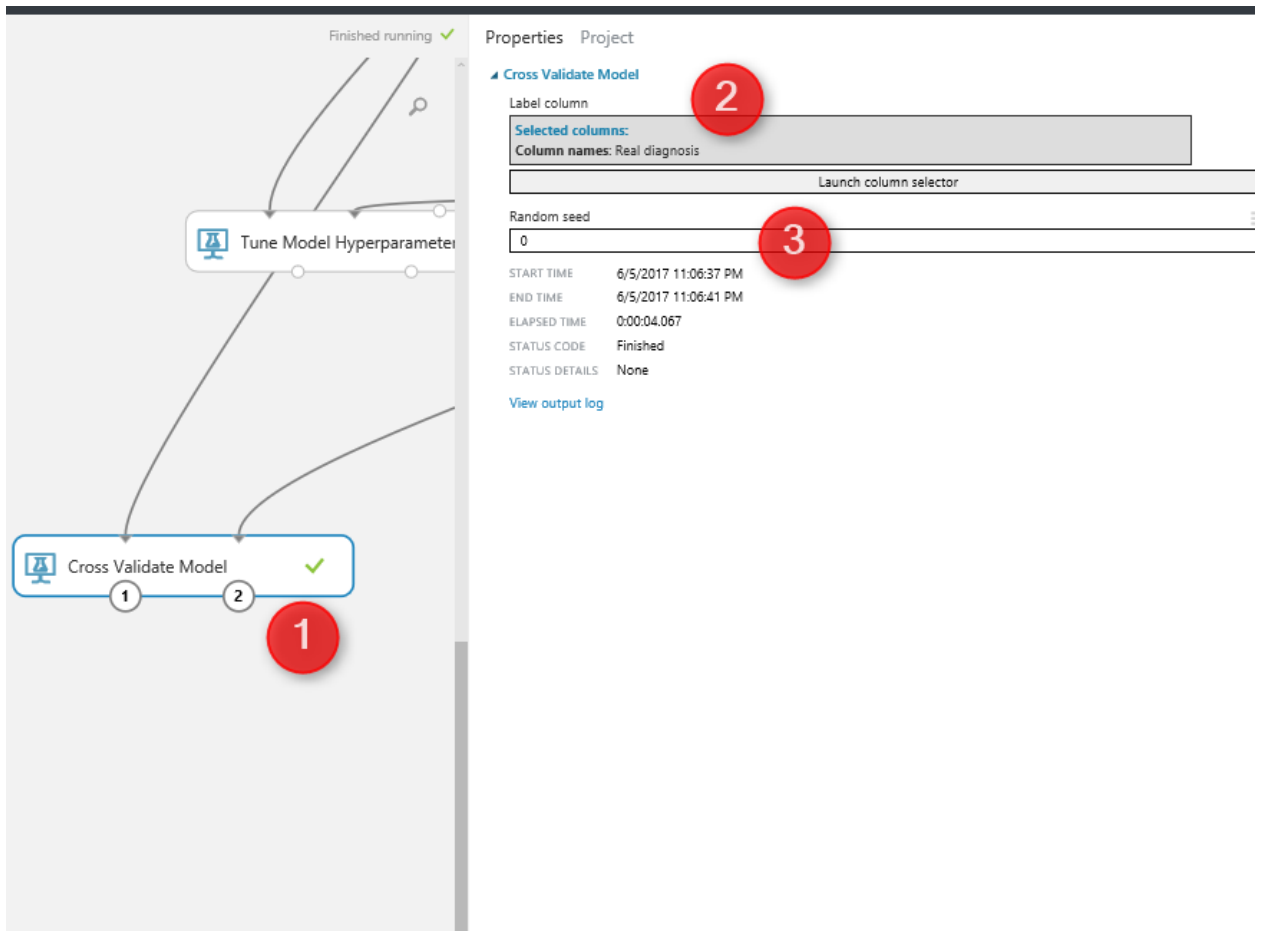
As you

can see in above picture, it gets one input from mode (decision forest) and the other input come from



the data for training.

By clicking on the “**Cross Validation Model**” component, we will see a properties panel that has just two parameters to set, the prediction column that is “real Diagnosis” and the Random



Finished running ✓

Tune Model Hyperparameter

Cross Validate Model ✓

1 2

1

Properties Project

Cross Validate Model

Label column

Selected columns:

Column names: Real diagnosis

Launch column selector

Random seed

0

3

START TIME 6/5/2017 11:06:37 PM

END TIME 6/5/2017 11:06:41 PM

ELAPSED TIME 0:00:04.067

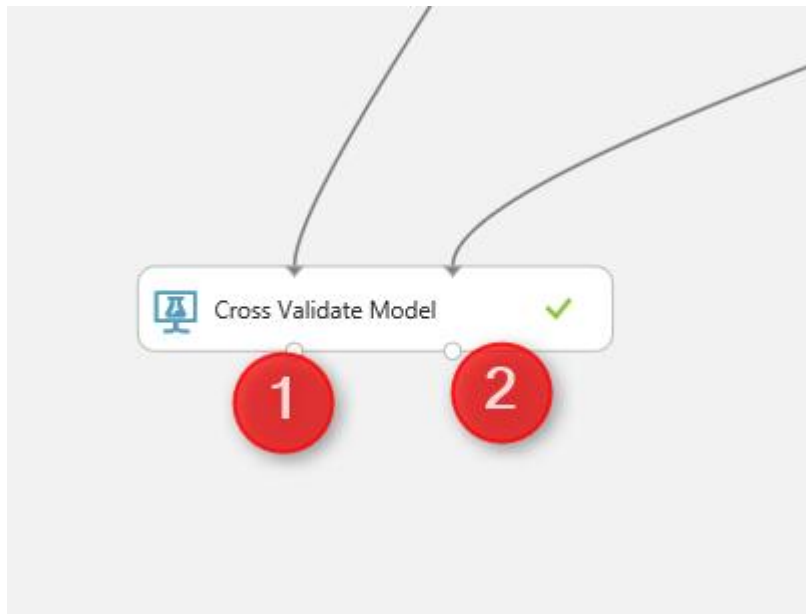
STATUS CODE Finished

STATUS DETAILS None

[View output log](#)

Seed.

We run the experiment, Then, by right click on the left side output node of Cross-Validation model (below picture number 1), and click on the visualize icon, you will see the result for



prediction.

The below windows will show up, as you can see in the first column, the fold number (dataset number) is in Column 1, and two last columns (column 13 and 14) show the prediction result and its probability to happen.

Cancer Prediction > Cross Validate Model > Scored results

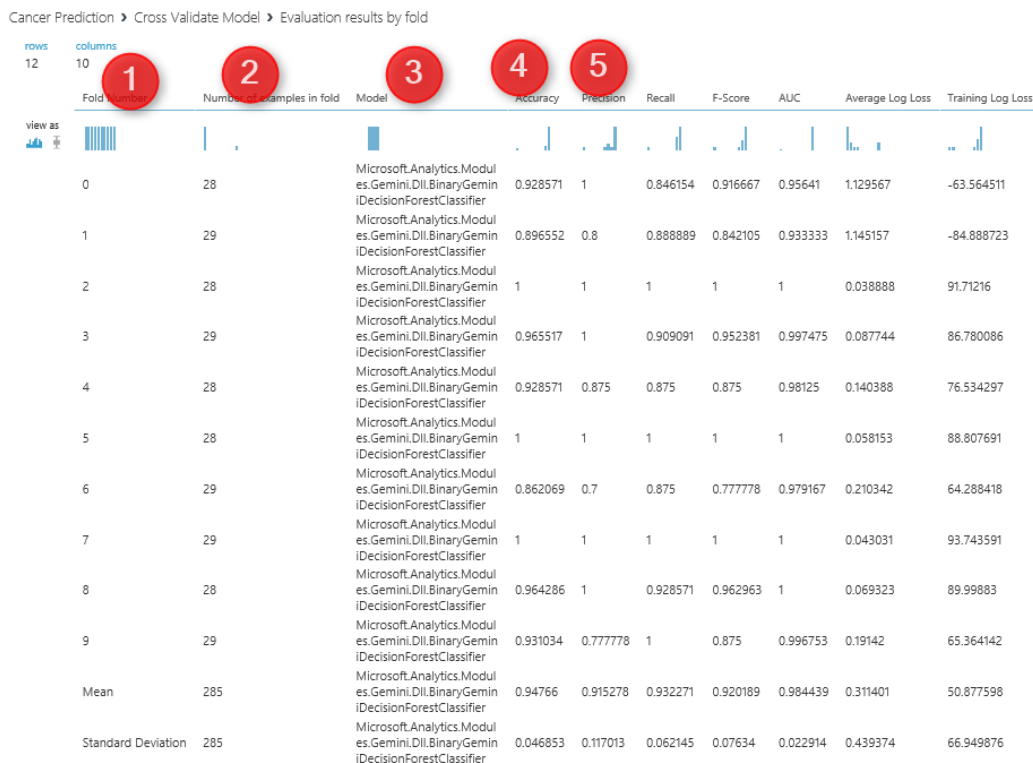
rows	columns																		
285	14																		
view as	view as	Fold Assignments	Real diagnosis	points_worst	perimeter_worst	points_mean	radius_worst	perimeter_mean	area_worst	radius_mean	area_mean	concavity_mean	concavity_worst	Scored Labels	Scored Probabilities				
4	Bar	Benign	0.136673	0.169929	0.04242	0.192814	0.235575	0.088257	0.251739	0.136119	0.018172	0.028906	Benign	0					
6	Bar	Benign	0	0.172668	0	0.196371	0.233396	0.091945	0.250319	0.136543	0	0	Benign	0					
6	Bar	Benign	0.379725	0.156598	0.223509	0.167556	0.218851	0.071997	0.219083	0.11228	0.166284	0.183866	Benign	0					
6	Bar	Benign	0.255361	0.147019	0.132326	0.159374	0.187478	0.070334	0.192106	0.097434	0.071368	0.085224	Benign	0					
8	Bar	Benign	0.088625	0.096867	0.064115	0.109925	0.143459	0.045075	0.15093	0.071432	0.025398	0.017316	Benign	0					
9	Bar	Benign	0.202405	0.239006	0.135835	0.270366	0.319536	0.138788	0.337404	0.200636	0.041354	0.042388	Benign	0.125					
7	Bar	Benign	0.164467	0.060511	0.142744	0.064141	0.103656	0.024381	0.090255	0.04263	0.20164	0.114537	Benign	0					
1	Bar	Benign	0.244536	0.172618	0.09667	0.183655	0.227904	0.083194	0.237541	0.126829	0.095009	0.144649	Benign	0					
0	Bar	Malignant	0.75945	0.235271	0.424602	0.254714	0.277659	0.128326	0.259312	0.149997	0.532568	0.882588	Benign	0.5					
7	Bar	Malignant	0.52268	0.276856	0.335934	0.287442	0.335429	0.148152	0.323678	0.191898	0.317948	0.270048	Malignant	1					
4	Bar	Benign	0.28134	0.204044	0.143241	0.192458	0.243867	0.088478	0.246533	0.132471	0.066893	0.079193	Benign	0					
6	Bar	Benign	0.182474	0.205289	0.092793	0.230167	0.268399	0.113203	0.27777	0.15737	0.143533	0.146805	Benign	0.125					
3	Bar	Benign	0.279038	0.10379	0.113917	0.10032	0.12335	0.041388	0.124237	0.058112	0.197329	0.346725	Benign	0.125					
3	Bar	Benign	0.503436	0.129987	0.295278	0.103878	0.182157	0.040997	0.169861	0.08263	0.534208	0.481629	Benign	0.25					
9	Bar	Benign	0.255361	0.170178	0.087972	0.192458	0.231014	0.089117	0.24606	0.133701	0.055834	0.091454	Benign	0					
4	Bar	Malignant	0.850515	0.273395	0.731113	0.782284	0.841061	0.603814	0.835297	0.720042	0.541237	0.372045	Malignant	1					
8	Bar	Benign	0.240103	0.215897	0.073211	0.241907	0.288093	0.117553	0.3038	0.173446	0.051101	0.108946	Benign	0					
3	Bar	Benign	0.14	0.101997	0.101243	0.104945	0.154861	0.042248	0.15519	0.075546	0.083903	0.052704	Benign	0					
8	Bar	Malignant	0.586942	0.577668	0.336581	0.548915	0.578467	0.369347	0.577358	0.426087	0.261012	0.308706	Malignant	1					
2	Bar	Benign	0.347079	0.1898	0.220676	0.198915	0.221823	0.090223	0.225709	0.118515	0.123758	0.109105	Benign	0					
9	Bar	Benign	0.428987	0.317695	0.175348	0.332266	0.352982	0.170959	0.368167	0.222778	0.107966	0.171805	Malignant	0.625					
0	Bar	Malignant	0.695533	0.492505	0.43494	0.43899	0.405017	0.266368	0.393724	0.249799	0.39433	0.50599	Malignant	1					
6	Bar	Malignant	0.520619	0.410827	0.523857	0.446105	0.508673	0.274479	0.506366	0.347911	0.43463	0.286182	Malignant	0.875					
9	Bar	Benign	0.348828	0.295284	0.305268	0.314123	0.379656	0.165086	0.381419	0.231559	0.180904	0.148243	Benign	0.25					
5	Bar	Malignant	0.717869	0.75696	0.700795	0.808965	0.769194	0.668698	0.78229	0.68017	0.540769	0.384904	Malignant	1					
3	Bar	Malignant	0.294433	0.326162	0.141501	0.355034	0.388432	0.197454	0.408396	0.260912	0.098618	0.180511	Malignant	0.625					
2	Bar	Benign	0.08811	0.062005	0.063718	0.068125	0.105176	0.027182	0.111316	0.051113	0.093627	0.063842	Benign	0					
6	Bar	Benign	0.517182	0.281837	0.387575	0.262184	0.373437	0.128736	0.344503	0.20632	0.703608	0.541773	Malignant	0.875					
7	Bar	Benign	0.195533	0.191344	0.075895	0.192458	0.238408	0.089019	0.245113	0.132259	0.068322	0.129393	Benign	0					
1	Bar	Malignant	0.685911	0.464117	0.498012	0.485237	0.519729	0.307658	0.502579	0.355037	0.500469	0.41254	Malignant	1					

By

clicking on the other output node of the cross-validated model, (number 2 in the previous

picture), you will see the analysis for each dataset (see below picture) so for first column (Fold) we have the dataset number or fold number, the second columns show the number of data in each fold. By default, we have ten folds, but you can change it using the “Partition and Sampling” component (will talk about it later in this chapter). The third column is about the model that we run. From column 4 to 10, you can see the accuracy measures that show the performance of the algorithms on each dataset. So as you look into accuracy in column 4, you will see that the value ranges from 89 to 1 which good and shows the dataset is pretty

Cancer Prediction > Cross Validate Model > Evaluation results by fold

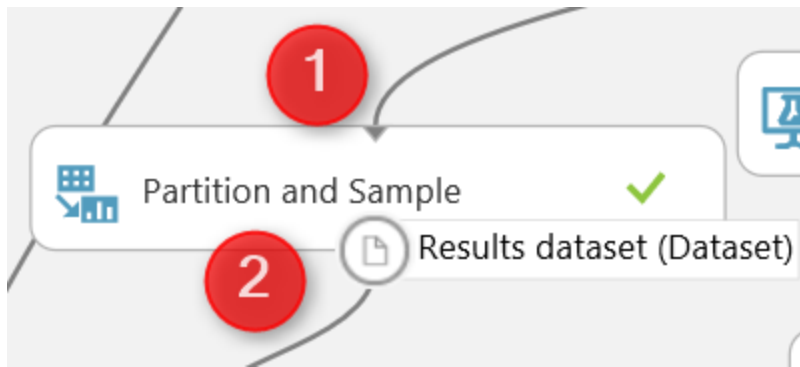


Fold	Number of Examples in fold	Model	Accuracy	Precision	Recall	F-Score	AUC	Average Log Loss	Training Log Loss
0	28	Microsoft.Analytics.Module.Gemini.Dll.BinaryGeminiDecisionForestClassifier	0.928571	1	0.846154	0.916667	0.95641	1.129567	-63.564511
1	29	Microsoft.Analytics.Module.Gemini.Dll.BinaryGeminiDecisionForestClassifier	0.896552	0.8	0.888889	0.842105	0.933333	1.145157	-84.888723
2	28	Microsoft.Analytics.Module.Gemini.Dll.BinaryGeminiDecisionForestClassifier	1	1	1	1	1	0.038888	91.71216
3	29	Microsoft.Analytics.Module.Gemini.Dll.BinaryGeminiDecisionForestClassifier	0.965517	1	0.909091	0.952381	0.997475	0.087744	86.780086
4	28	Microsoft.Analytics.Module.Gemini.Dll.BinaryGeminiDecisionForestClassifier	0.928571	0.875	0.875	0.875	0.98125	0.140388	76.534297
5	28	Microsoft.Analytics.Module.Gemini.Dll.BinaryGeminiDecisionForestClassifier	1	1	1	1	1	0.058153	88.807691
6	29	Microsoft.Analytics.Module.Gemini.Dll.BinaryGeminiDecisionForestClassifier	0.862069	0.7	0.875	0.777778	0.979167	0.210342	64.288418
7	29	Microsoft.Analytics.Module.Gemini.Dll.BinaryGeminiDecisionForestClassifier	1	1	1	1	1	0.043031	93.743591
8	28	Microsoft.Analytics.Module.Gemini.Dll.BinaryGeminiDecisionForestClassifier	0.964286	1	0.928571	0.962963	1	0.069323	89.99883
9	29	Microsoft.Analytics.Module.Gemini.Dll.BinaryGeminiDecisionForestClassifier	0.931034	0.777778	1	0.875	0.996753	0.19142	65.364142
Mean	285	Microsoft.Analytics.Module.Gemini.Dll.BinaryGeminiDecisionForestClassifier	0.94766	0.915278	0.932271	0.920189	0.984439	0.311401	50.877598
Standard Deviation	285	Microsoft.Analytics.Module.Gemini.Dll.BinaryGeminiDecisionForestClassifier	0.046853	0.117013	0.062145	0.07634	0.022914	0.439374	66.949876

well.

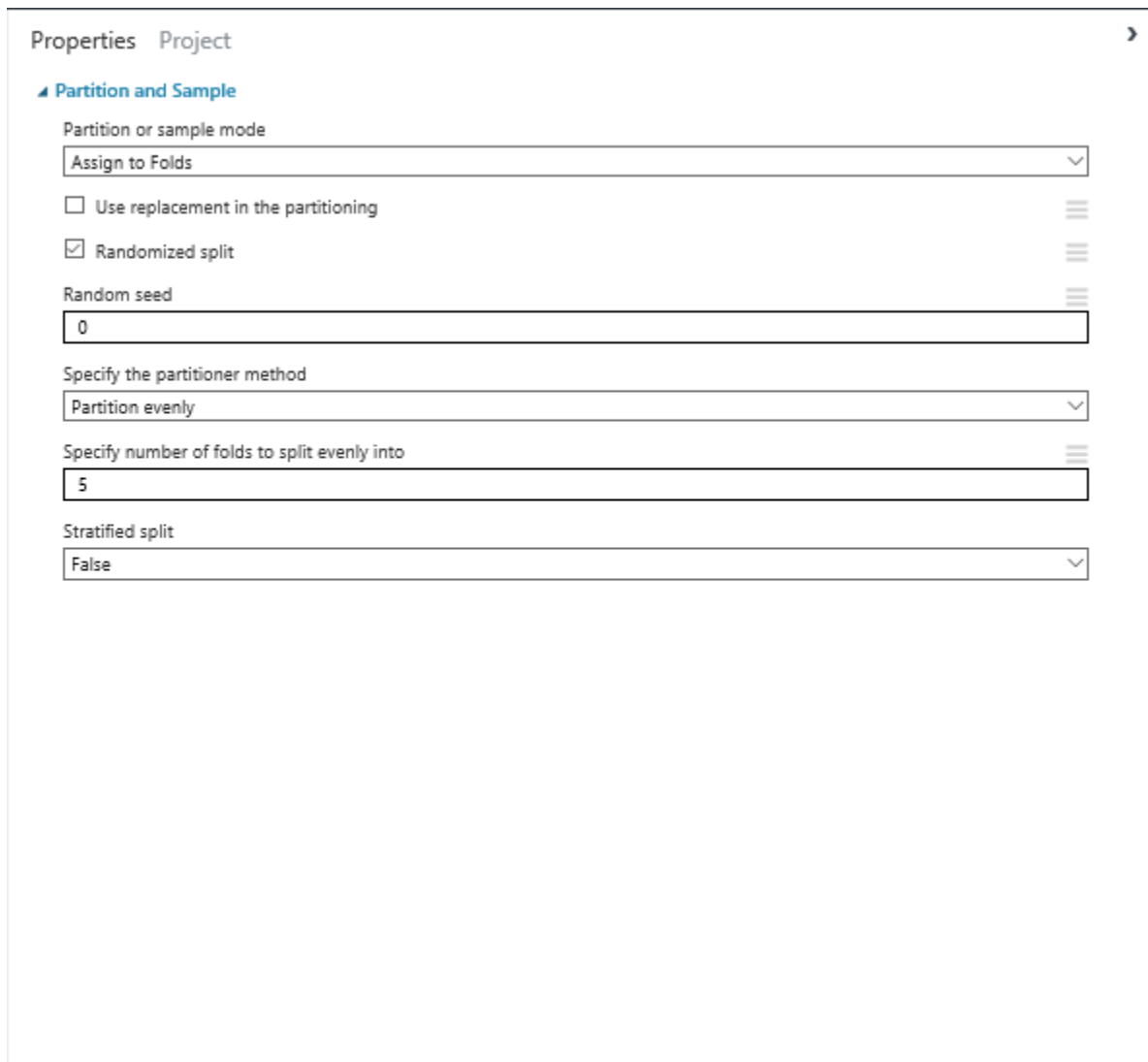
if

you wish to specify the number of folds yourself, then you have to use the component name “**Partition and Sample**”. See the below picture. This component gets one input from the split dataset and one output that has the data for cross-validation



if you click on the component, you will

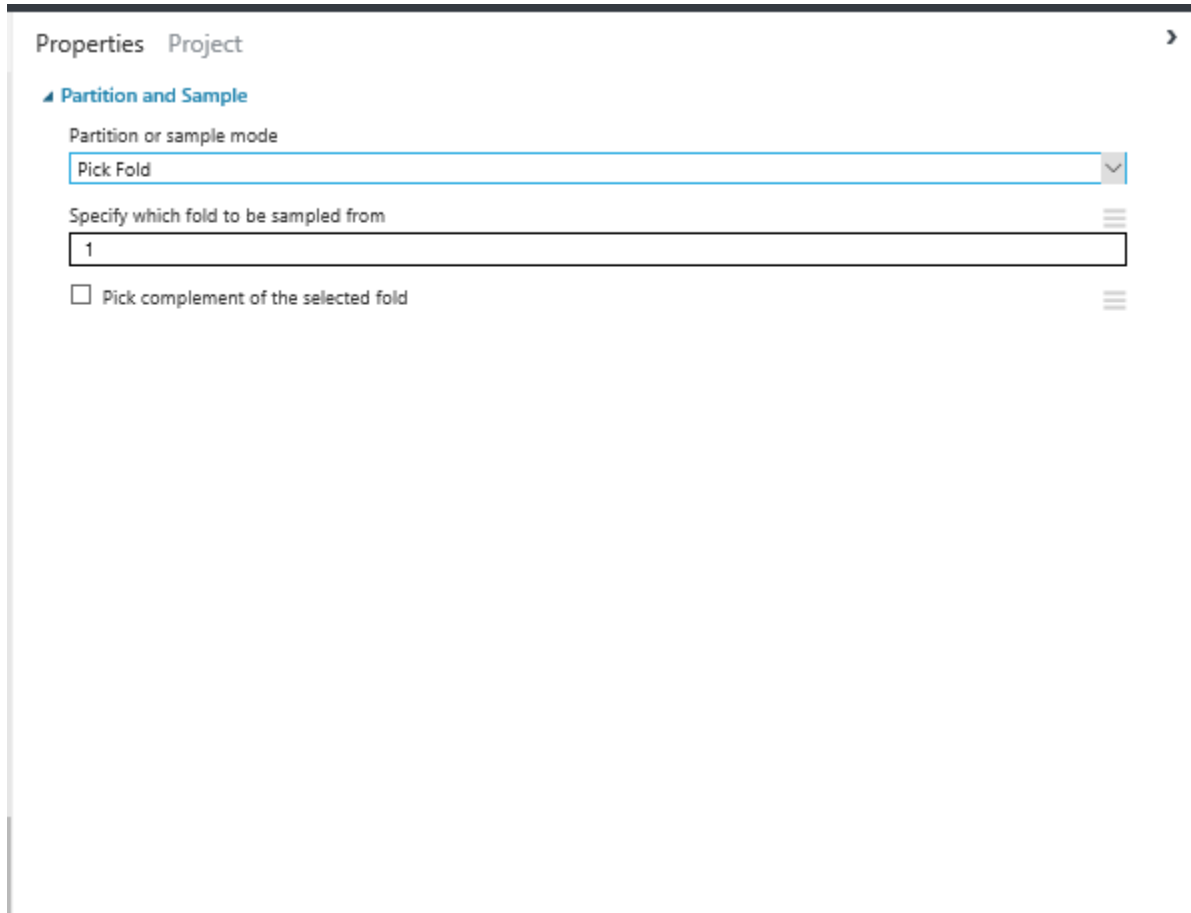
see the properties panel in the right side.



As you see in the above picture, there are some parameters that we able to assign. First for Cross-validation, I have chosen the **“Assign to Folds”** value for partition and sample mode. Also I have ticked the

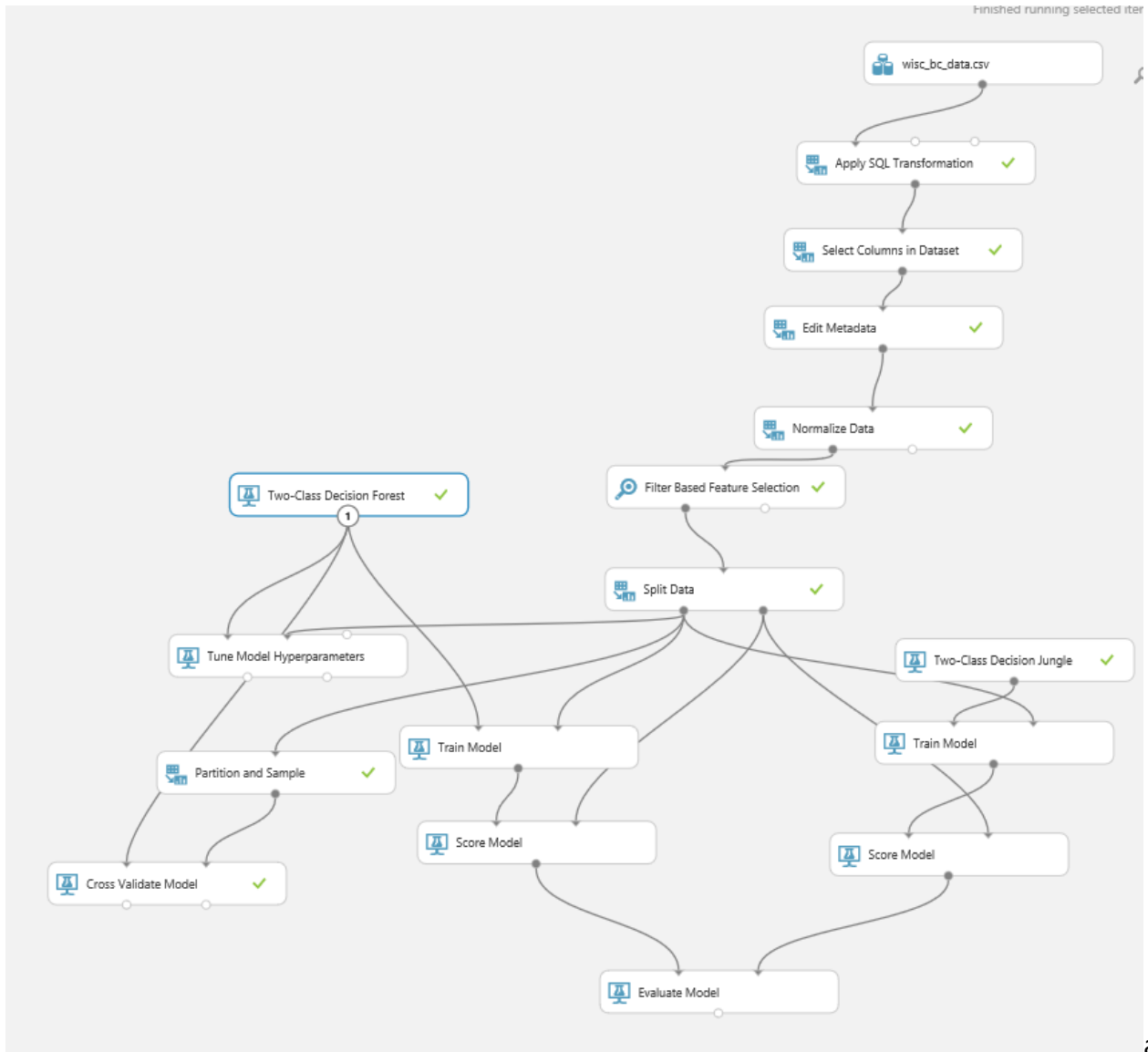


option for **“Random Split”** to decrease the possibilities of biased data selection. And finally for Specify number of folds to split I choose 5. so instead of 10 folds of data in cross-validation, now we have five folds. Also there is a possibility to pass one data fold to cross-validation by selecting **“Pick Fold”** as you can see in below picture



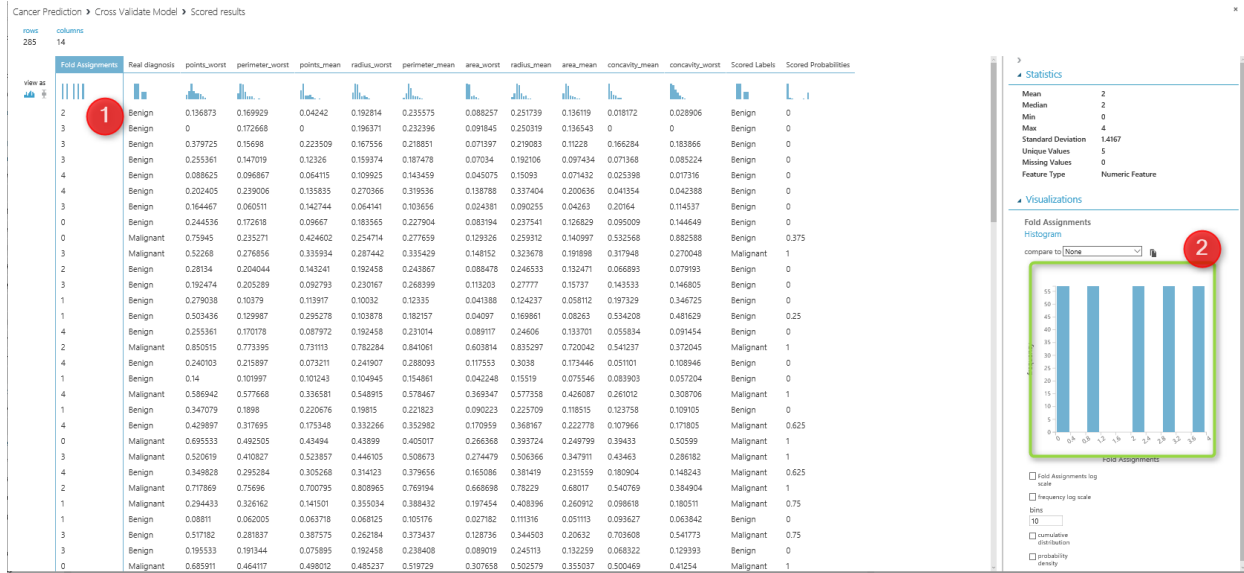
so we have

below process



as you

can see in above picture, I have connected the output of the partition and sample to the input of the cross-validation method. And I run the experiment, so I saw below data. I click on the first column (fold assignment), and on the right side of the windows, There is a chart that shows the summary of data. You see there that we have five folds now instead of 10.



In the

next Chapters, I will talk about the creating web service from Azure ML Studio model, how we can use it in Excel or other application. [https://en.wikipedia.org/wiki/Cross-validation\\_\(statistics\)](https://en.wikipedia.org/wiki/Cross-validation_(statistics))  
<https://msdn.microsoft.com/library/azure/75fb875d-6b86-4d46-8bcc-74261ade5826>

# Chapter 10: Create Web service from Models

Published Date: February 27, 2019

In previous [Chapters](#), I have explained how to create a machine learning scenario using Azure ML Studio components.

One of the main advantages of using Azure ML Studio is the ability to create a web service from it. That means the created model can be used in other services via an API URL and Password.

In this chapter, I will show how to create an API for a predictive model.

In this chapter, we are going to use Titanic dataset for the aim of creating a web service.

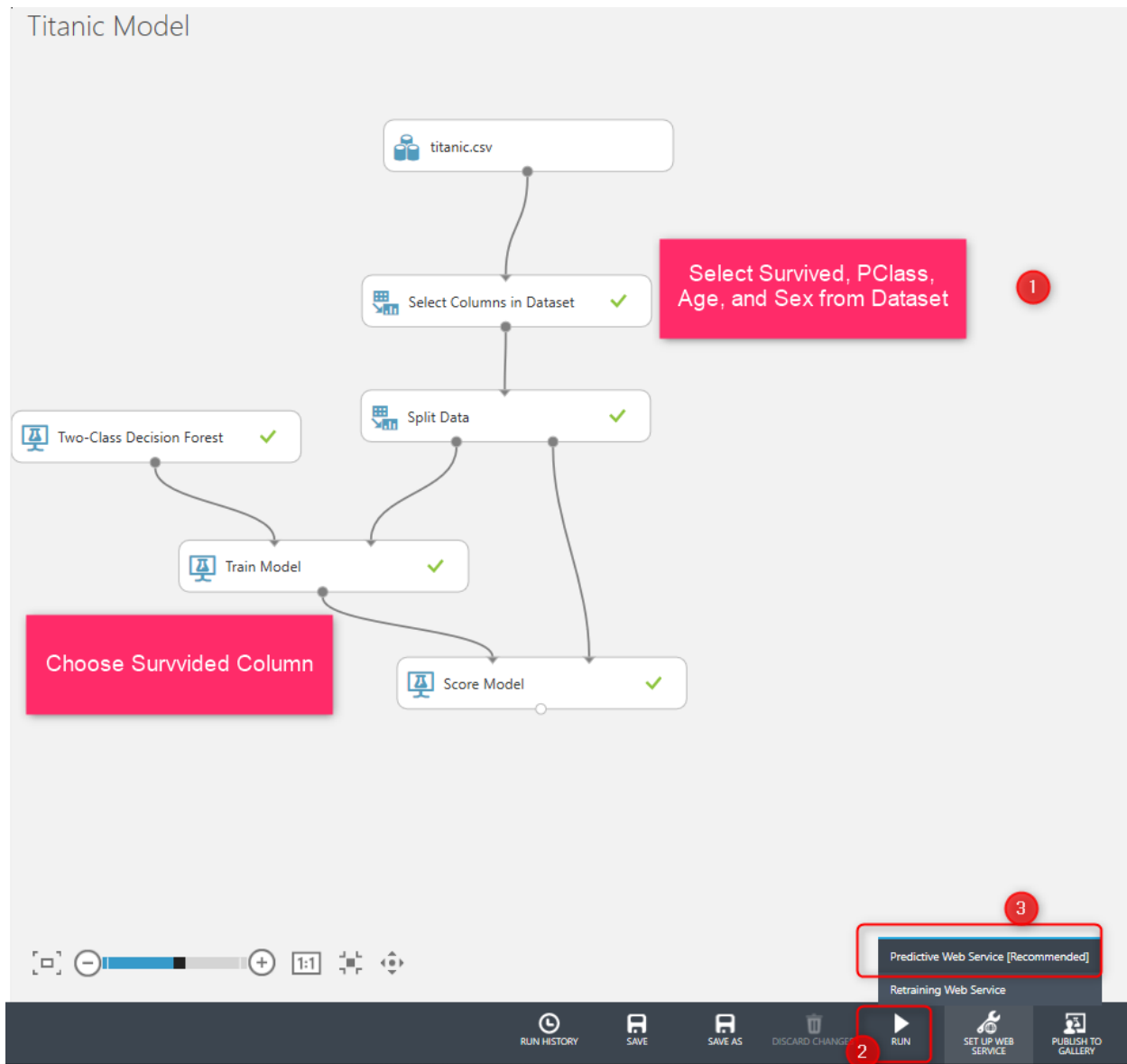
We have a dataset about Titanic, that has information such as passenger class, age, gender and people survived or not. Also, there is some other information such as passenger name and ID exists which we do not need them at this stage.

To access the Titanic Dataset, you can find it in all machine learning weblog and website such as Kaggle [1].

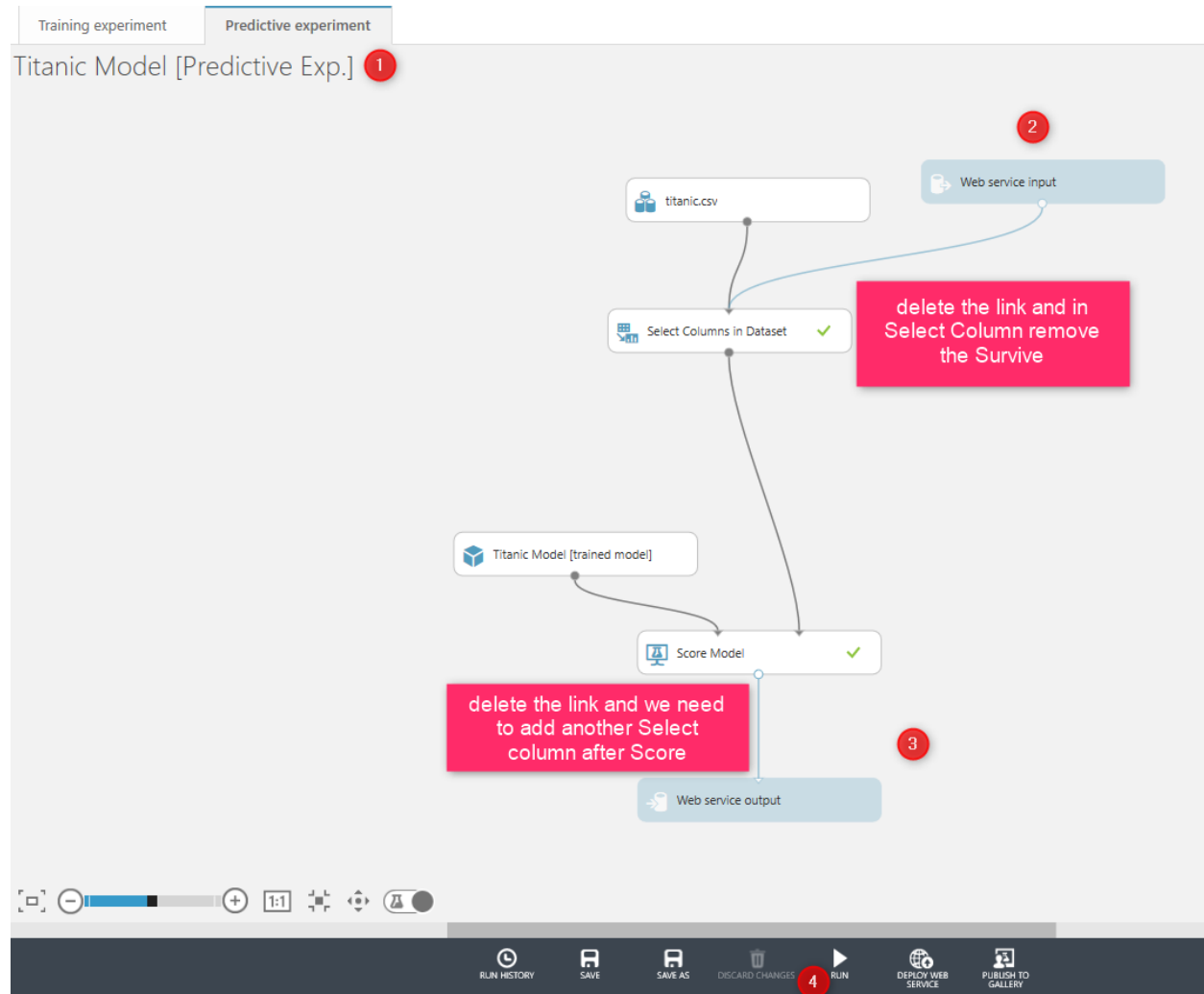
Import the dataset and create the below model in Azure ML Studio Studio Experiment.



As you can see in the above picture, there is a need to select only Survived, Age, Sex, and PClass from all column list and for train model just Survived Column.



after creating the model, you need to run it, then click on the *Predictive Web Service* to access the web service. Next, (as you can see in the below picture, two different tabs exists one for experiment and the other one for Predictive experiment. And we have two extra nodes one for the input web service and another one for the output web service.

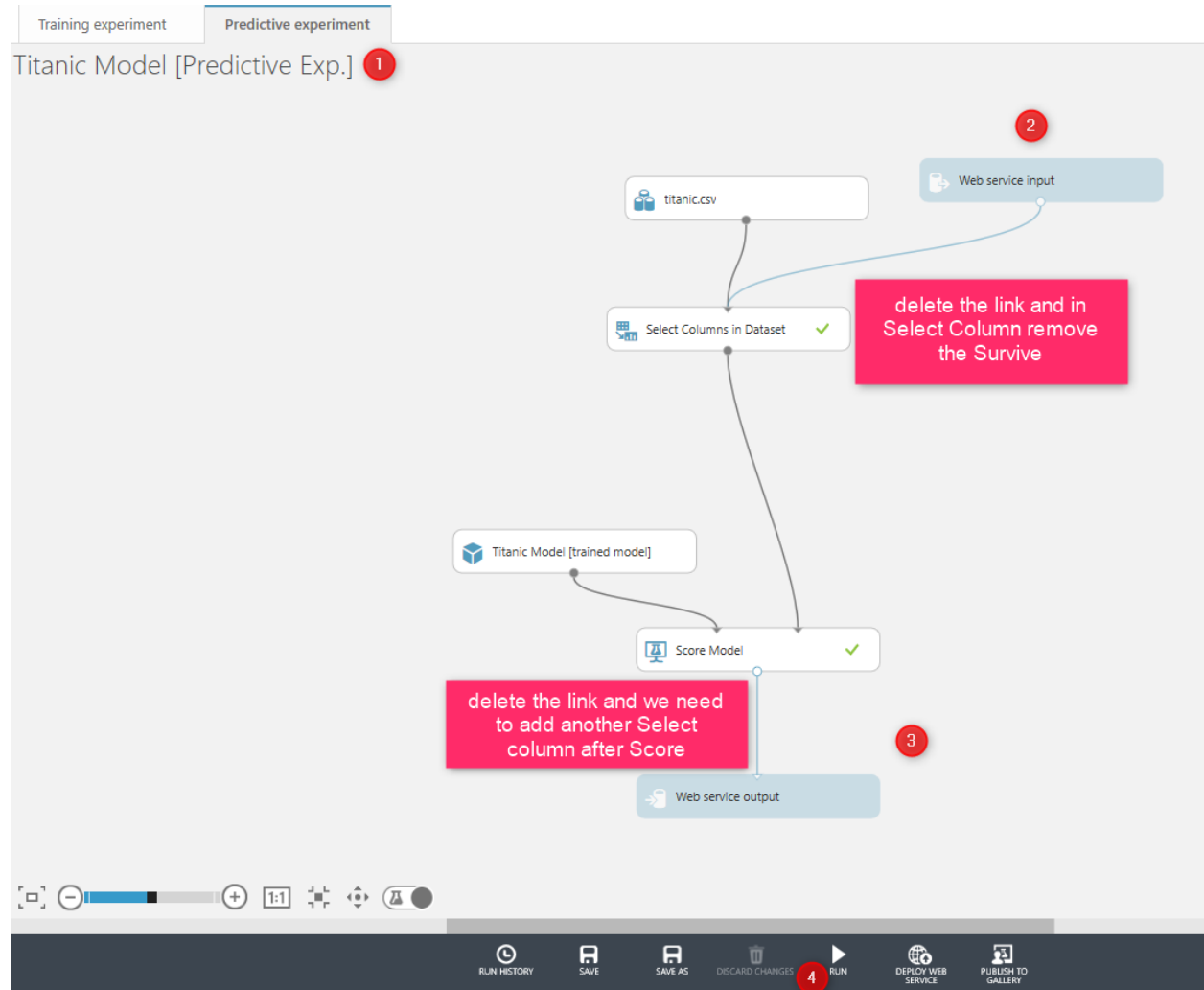


run the experiment, however, there is a need to change the input and output of the web services.

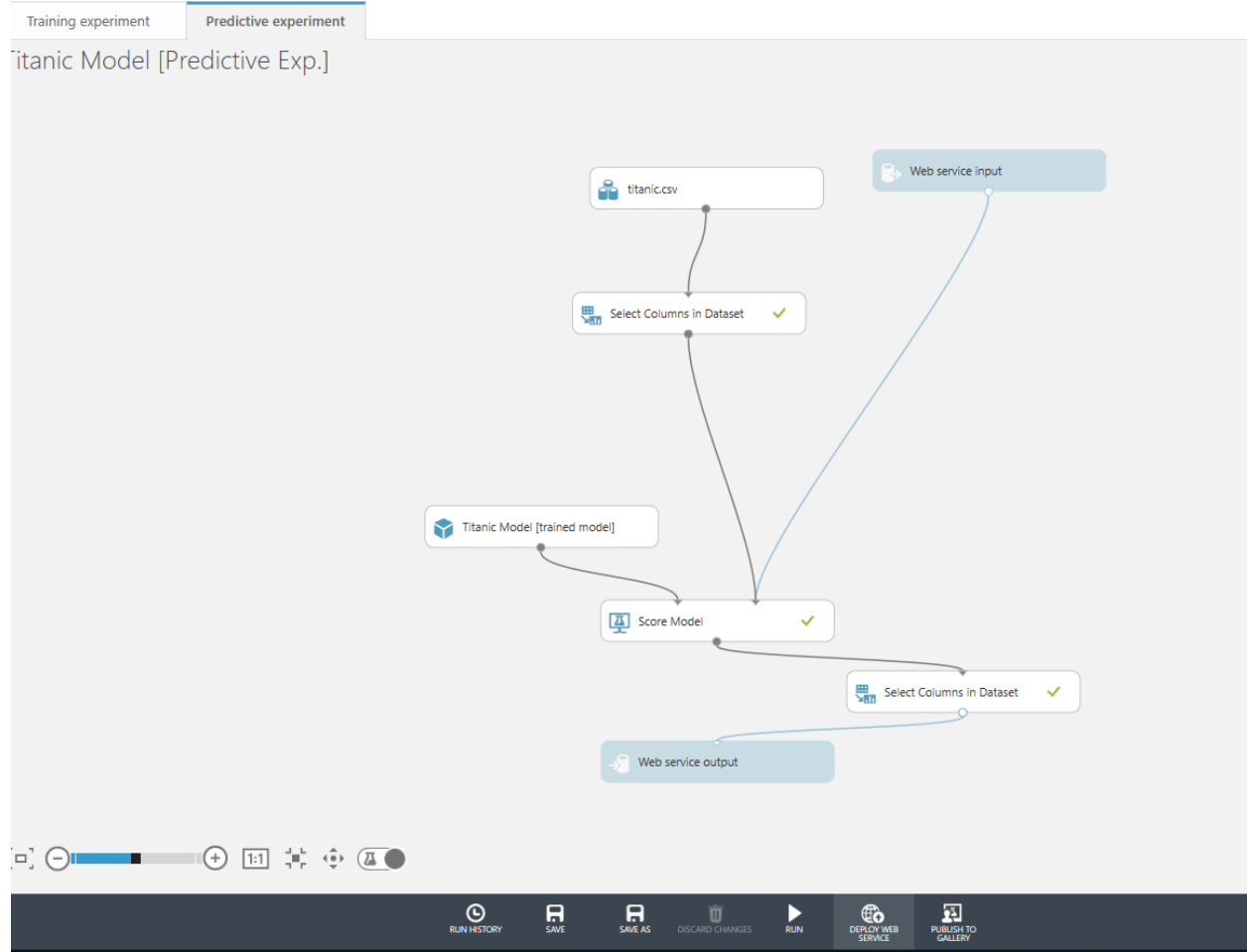
Input Web service needs to get the input from users that is *Age, Gender, and their PClass*. For the output is the same, end users need to see the prediction results which is *Score and Score Probabilities*.

As a result, we need to change the model before creating the web service, remove the connection between input web service and connect it to the input of the Score model, then customized the select column node and remove the Survived column from the list (because for the input of the web service, user just need to provide age, gender, and passenger class, not the survived condition)

Next, you need to modify the connection for the output as well, so delete the current connection between the output node and, then add a new Column Selector node which selects only Score and Score Probability. Then connect the input of the output web service to it (as shown in the below picture). Then run the experiment again and click on the deploy web service at the bottom of the page



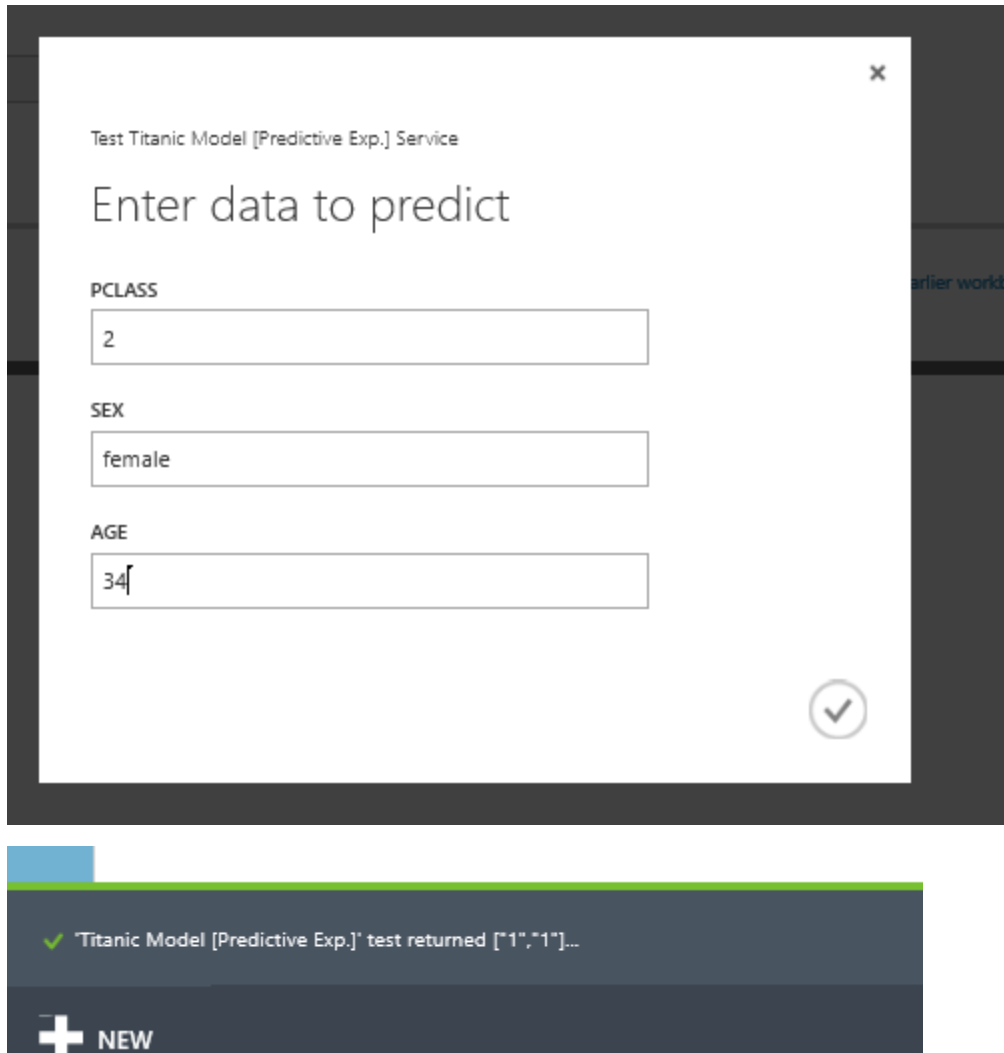




Now, a new page will show that provide some detail information about the web service.

In the new page you able to see the API code, the request and response link that shows how to set up web service and also the Test button to see how web service work.

click on the Test and see the result



Test Titanic Model [Predictive Exp.] Service

Enter data to predict

PCLASS

2

SEX

female

AGE

34

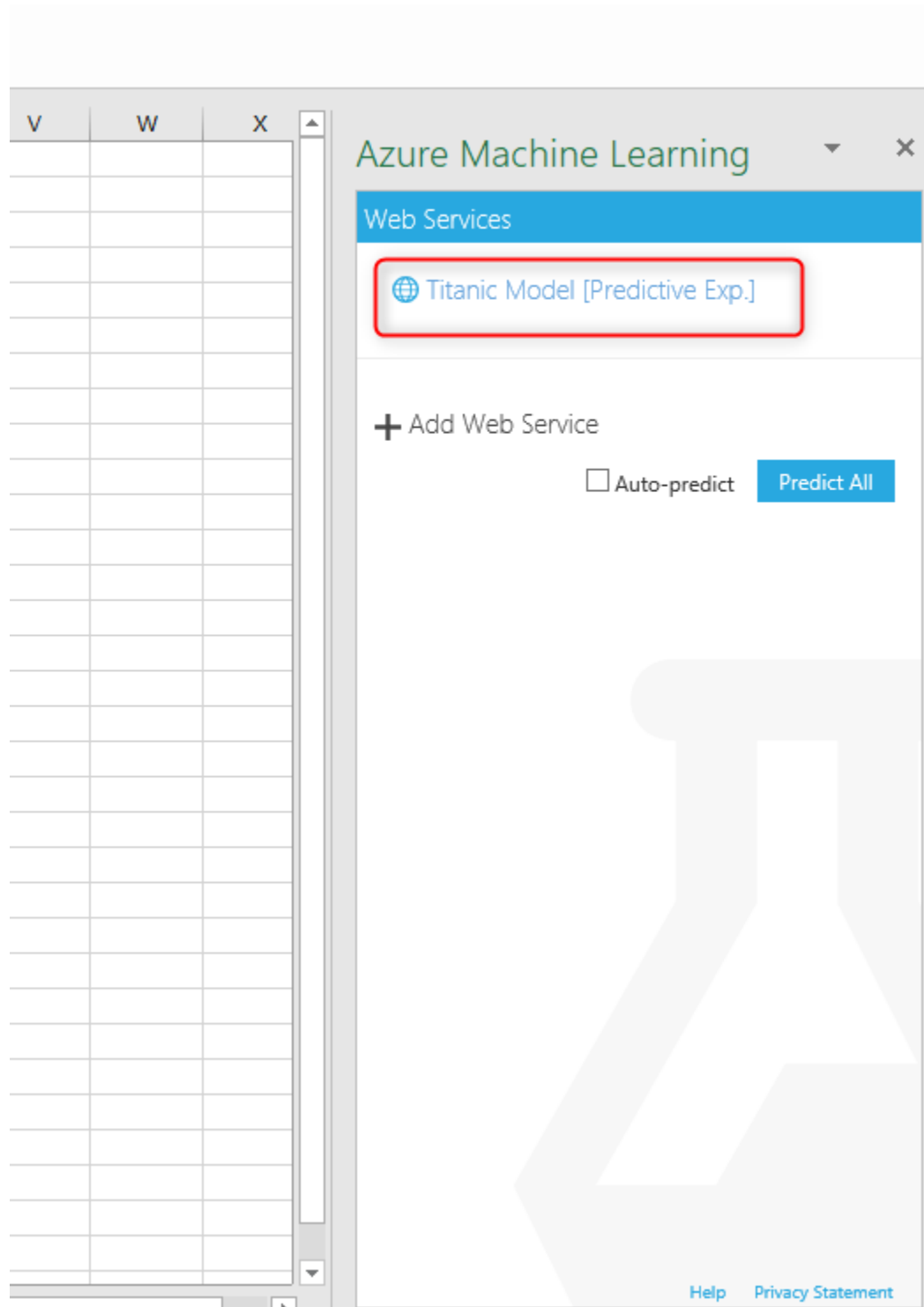
✓ Titanic Model [Predictive Exp.] test returned ["1", "1"]...

+ NEW

as you can see in the above picture, you need to provide some information and at the bottom of the page the result will show up.

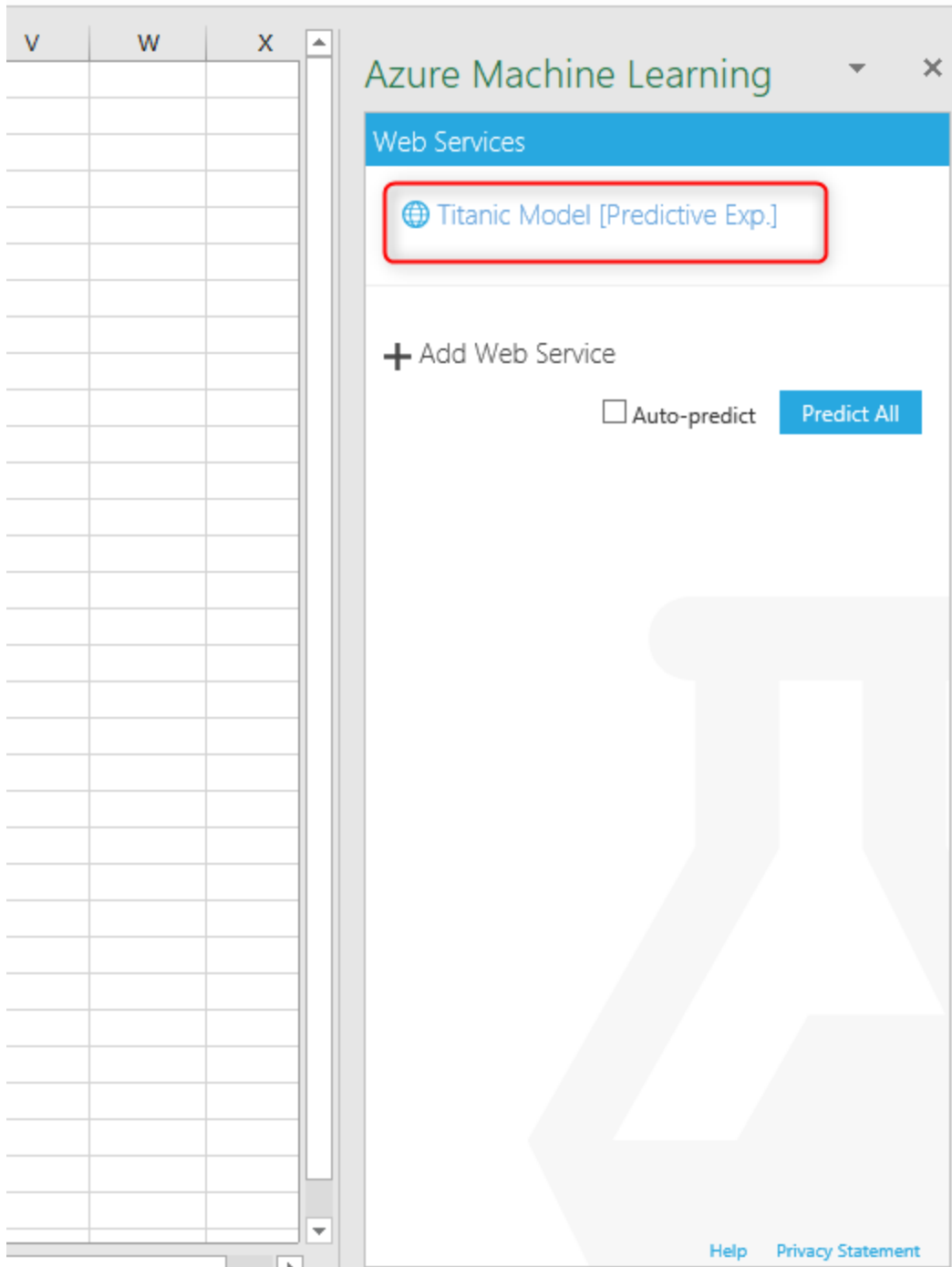
In the next step, you able to analyze the data in Excel by clicking on the *Excel 2013 or later*. By clicking a new Excel file with sample data will be download, enable the file and then click on the web service name

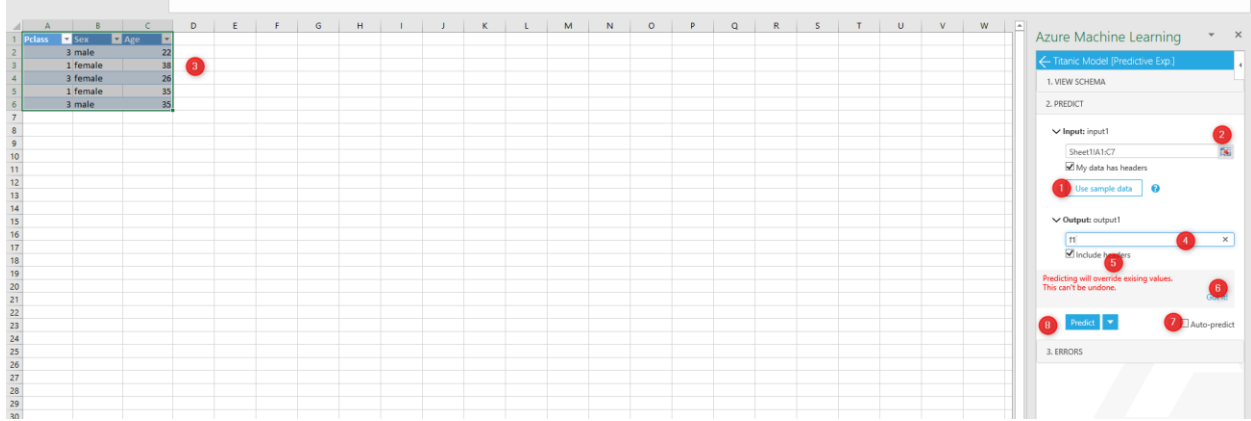
APPS	TEST	LAST UPDATED
REQUEST/RESPONSE	Test Test preview	2/27/2019 9:02:54 AM
BATCH EXECUTION	Test preview	2/27/2019 9:02:54 AM



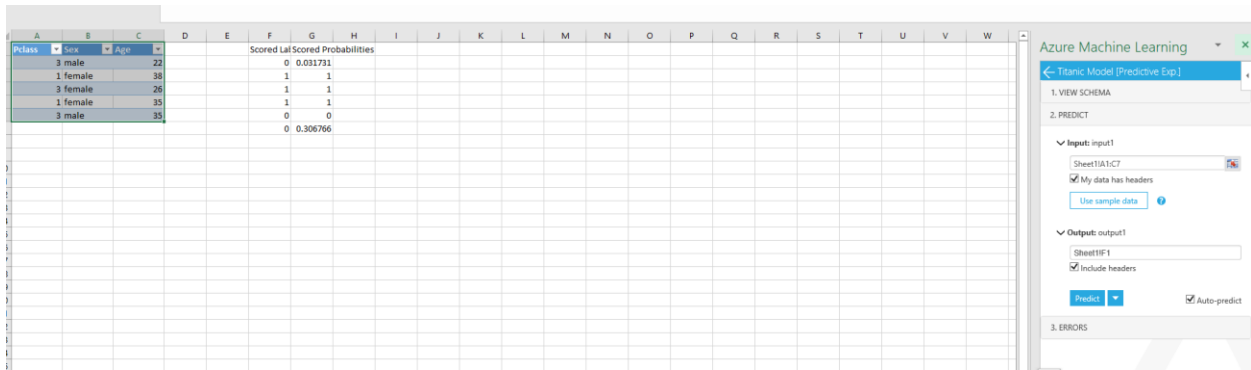
The screenshot shows the Azure Machine Learning Studio interface. On the left is a grid with columns labeled 'V', 'W', and 'X'. The main area is titled 'Azure Machine Learning' and contains a 'Web Services' section. A red box highlights a service named 'Titanic Model [Predictive Exp.]'. Below this, there is an '+ Add Web Service' button, an 'Auto-predict' checkbox, and a 'Predict All' button. At the bottom right, there are links for 'Help' and 'Privacy Statement'.

select the *Use Sample Data*, then under the input 1, choose the data range, then identify the output range click on the *Got it, Auto-predict and then Predict option to see the results.*





Pclass	Sex	Age
3	male	22
1	female	38
3	female	26
1	female	35
3	male	35



Pclass	Sex	Age	Scored Lat Scored Probabilities
3	male	22	0 0.031731
1	female	38	1 1
3	female	26	1 1
1	female	35	1 1
3	male	35	0 0
			0 0.306766

[1] Kaggle: <https://www.kaggle.com/francksylla/titanic-machine-learning-from-disaster/data>

# Upcoming Training Courses

Leila runs different Microsoft Machine Learning training courses both online and in person. RADACAD also runs Power BI and SQL Server courses ran by Reza Rad. Our courses run both online and in person in major cities and countries around the world. Check the schedule of upcoming courses here:

<http://radacad.com/events>

<http://radacad.com/advanced-analytics-training>