

Classification in Azure By Nan(Miya) Wang

Data:

In this lesson we use the dataset *churn.txt*, containing information on 1,477 customers of a telecommunication firm who has at some time purchased a mobile phone. The customers fall into one of three groups: current customers, involuntary leavers, and voluntary leavers. The file contains information on the customer account including length of time spent on local, long distance and international calls, the type of billing scheme, and a variety of basic demographics, such as age and gender. The data are typical of what is often referred to as a churn example (hence the file name).

The original dataset uses comma-separated txt format. **Machine Learning Studio works better with a comma-separated value (CSV) file, so we'll first convert the dataset into CSV file.**

There are many ways to convert this data. One way is by using the following Windows PowerShell command:

```
cat churn.txt | sc churn.csv
```

If you are using Unix, here are the commands:

```
churn.txt > churn.csv
```

In either case, we have created a comma-separated version of the data in a file named **churn.csv** that we'll use in our experiment.

Introduction

We will use the data file *churn.txt*. We will attempt to find natural segments, or clusters of customers to see whether they can be targeted for different promotions and to explore if cluster differences relate to customer status.

Steps:

Prepare the Data

1. Open a browser and browse to <http://studio.azureml.net>
2. Then sign in using the Microsoft account associated with your Azure ML account (or your school account if available)
3. Create a new dataset named churn.csv by clicking 'NEW' in the button, then clicking 'DATASET' and uploading the churn.csv file, as shown in the screenshot below. Click the little check in the circle in the right corner to continue.

×

Upload a new dataset

SELECT THE DATA TO UPLOAD:

C:\Users\Miya\OneDrive\DobinGA\churn.csv

Browse...

☐

This is the new version of an existing dataset

ENTER A NAME FOR THE NEW DATASET:

churn.csv

SELECT A TYPE FOR THE NEW DATASET:

Generic CSV File with a header (.csv)

▼

PROVIDE AN OPTIONAL DESCRIPTION:

✓

4. Click ‘NEW’ again and create an **EXPERIMENT**. In this stage, just click ‘**Blank Experiment**’.

A blank experiment canvas would appear.

5. Search for ‘churn.csv’ in the searching column on the left side and drag it onto the blank canvas.

6. Right-click on the **circle** on churn module and choose **visualize**. Now you can form a general idea of this dataset. Click and go through every data column, statistical facts and visualization of this column automatically formed in the right side.

rows

1477

columns

15

	ID	LONGDIST	International	LOCAL	DROPPED	PAY_MTHD	LocalBillType	LongDistanceBill
view as								
	0	5.2464	7.5151	86.3278	0	CH	FreeLocal	Standard
	3	0	0	3.94229	0	CC	Budget	Intl_discount
	4	5.55564	0	9.36347	1	CC	Budget	Intl_discount
	8	14.0193	5.68043	29.8065	0	CC	Budget	Standard
	10	13.664	2.95642	32.6381	0	CC	FreeLocal	Intl_discount
	11	0	0	1.41294	0	CC	FreeLocal	Standard
	13	0.281029	0	8.53692	0	CH	Budget	Intl_discount
	17	1.577	0	19.9808	0	CC	FreeLocal	Standard
	19	11.0307	0	34.2777	0	CC	Budget	Standard
	20	0.452629	0	73.0122	0	Auto	FreeLocal	Standard

Statistics

Mean	13.6373
Median	13.683
Min	0
Max	29.982
Standard Deviation	9.3942
Unique Values	1310
Missing Values	0
Feature Type	Numeric Feature

Visualizations

LONGDIST

Histogram

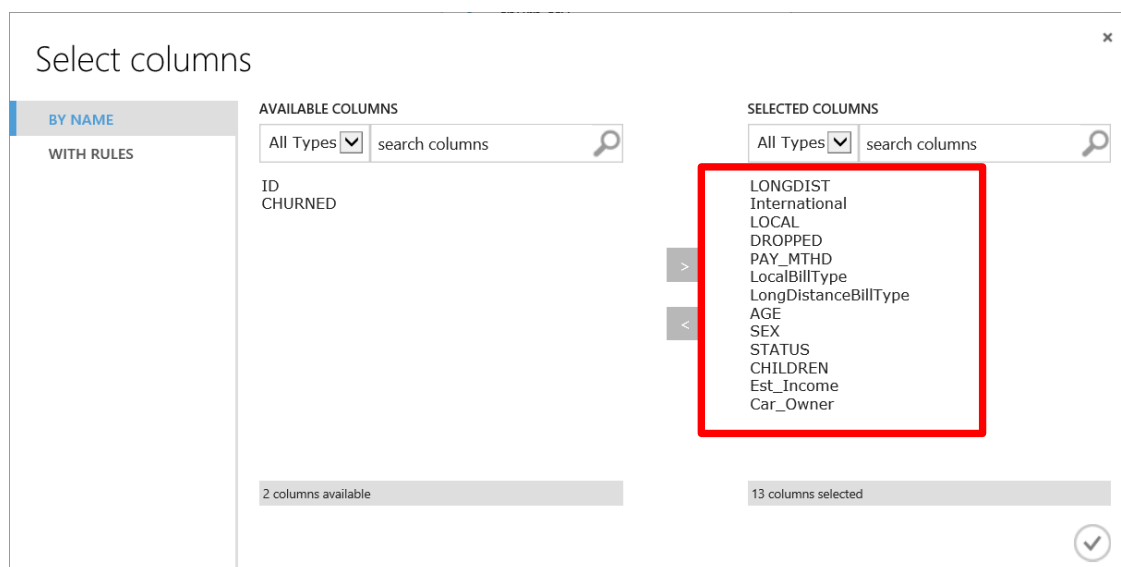
compare to None  

You may notice many column values labelled as string. We need to convert them into categorical features later when training the model.

7. Search for '**Metadata Editor**' and drag it onto the canvas. Connect it to churn dataset by clicking on the circle of the churn module(not letting go of your mouse) and directing the link to the upside circle on the 'Project Columns' module.

8. Click on 'Project Columns' and then click on '**Launch column selector**' on the right.

9. Select all the columns **except ID and Churned** in the dataset (as shown below). Click on the check circle in the right corner to continue.



10. Turn to 'Properties' on the right, and change the value of Fields into 'Features'.

Properties

Metadata Editor

Column

Selected columns:
Column names:
LONGDIST, International, I

Launch column selector

Data type

Unchanged

Categorical

Unchanged

Fields

Features

New column names

Quick Help

11. Search for 'Metadata Editor' again and drag it onto the canvas. Connect it to metadata above. This time, select columns **except ID, LONGDIST, International, LOCAL, CHILDREN, Est_Income**, as shown below.

Select columns

BY NAME

WITH RULES

AVAILABLE COLUMNS

All Types search columns

ID
LONGDIST
International
LOCAL
CHILDREN
Est_Income

6 columns available

SELECTED COLUMNS

All Types search columns

LocalBillType
LongDistanceBillType
DROPPED
PAY_MTHD
CHURNED
AGE
SEX
STATUS
Car_Owner

9 columns selected

✓

12. Turn to **‘Properties’** on the right, and change the value of Categorical into **‘Make Categorical’**.

Properties

Metadata Editor

Column

Selected columns:
Column names:
LocalBillType, LongDistance

Launch column selector

Data type

Unchanged

Categorical

Make categorical

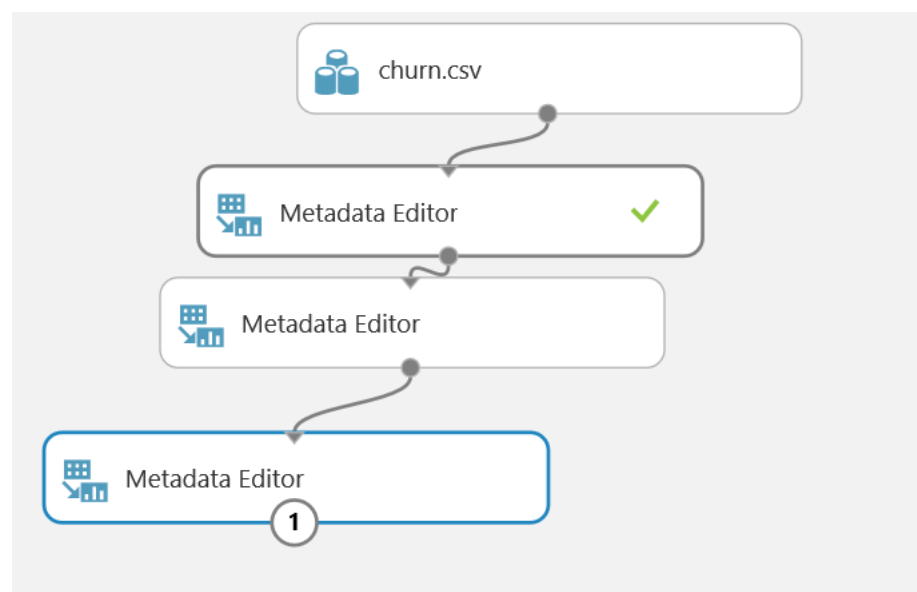
Fields

Unchanged

New column names

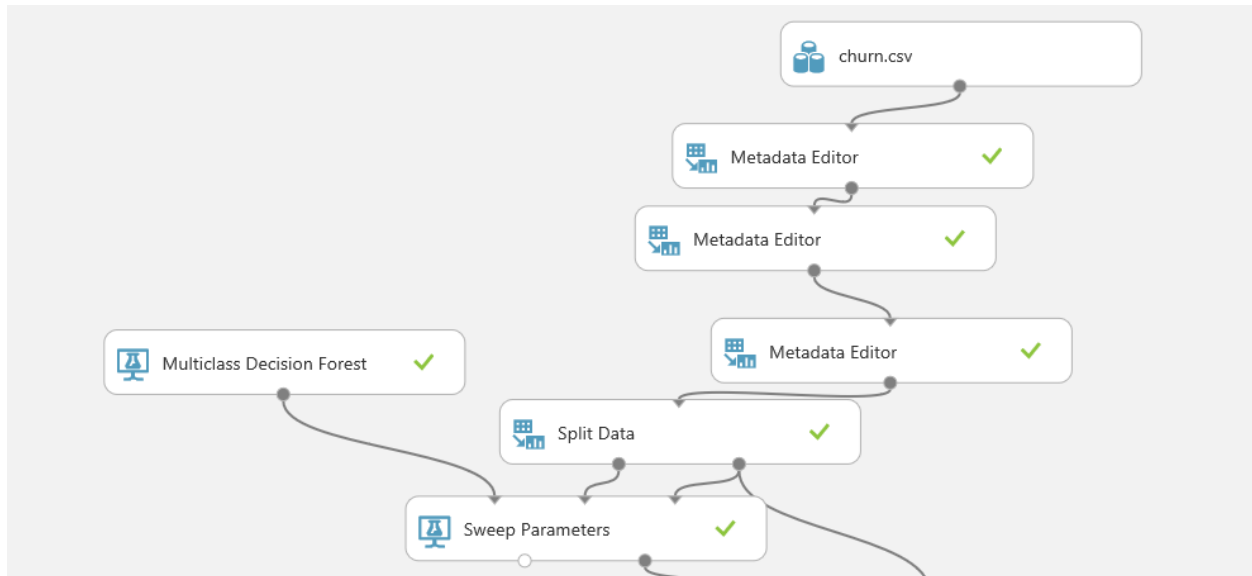
13. Search for **‘Metadata Editor’** again and drag it onto the canvas. Connect it to metadata above. This time, select only **Churned** column and change values of Fields into **‘label’**.

Now your experiment should look like this, with all data prepared for model building.



14. Search for 'Split Data' and drag this module onto the experiment canvas. Connect it to the last Metadata Editor Module. You can leave the default settings of this module except setting a random number for seed.

15. Search for '**Sweep Parameters**' module and multi-class **Decision Forest**. Make connections between them like the following screenshot.



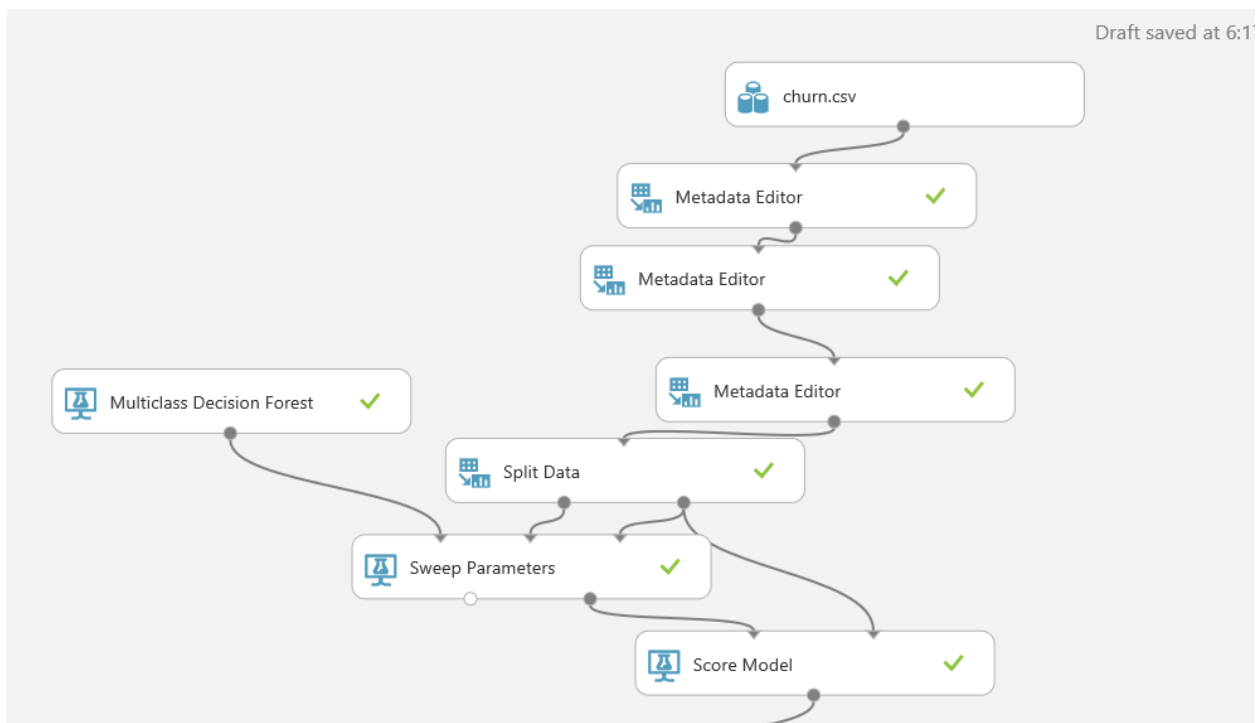
Notes: Sweep parameters is very useful for Data Mining beginners for it would help you set parameters in any algorithms you choose.

Azure has many choices of algorithms for multi-class classification. For example, multi-class neural network and multiclass logistic regression. Since application process is similar, you can play all of them around using the same steps we are talking about here!

16. Click on the 'Sweep Parameters' and click on 'Launch column selector'. Choose **Churned** only as label column.

17. You can customize your score metric by changing **metric for measuring performance on classification**. Here we just leave the default choice of **accuracy** unchanged.

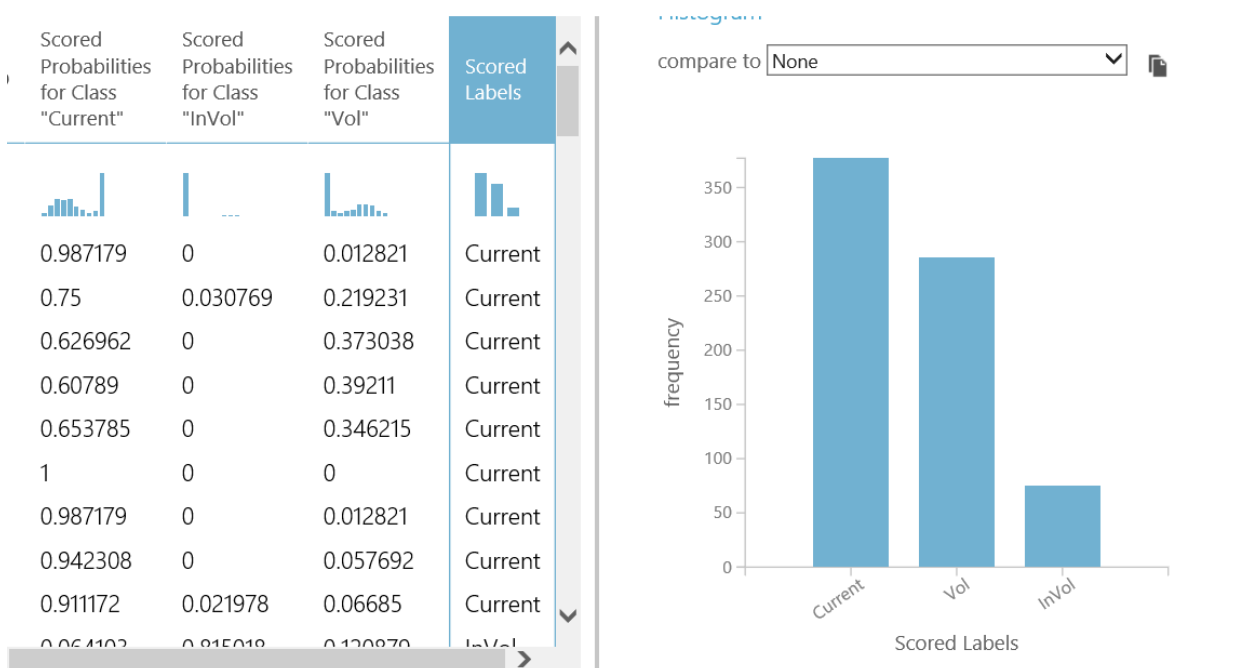
18. Search for '**Score Model**' and make connections like the screenshot showing below.



19. Now click on the **Run** button under the experiment canvas.

20. When finished running, click on the circle of the score model module and choose Visualize. You can see extra columns added to original dataset: **Scored probabilities for each class and scored labels**.

You can also see distributions of labels in the right.

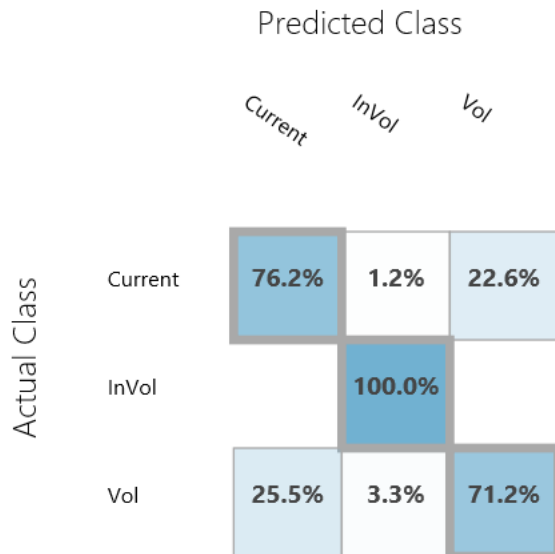


21. Now search for ‘**Evaluate Model**’ and drag it to the canvas. Make it connected to Module ‘**Score Model**’. Then run the experiment again.

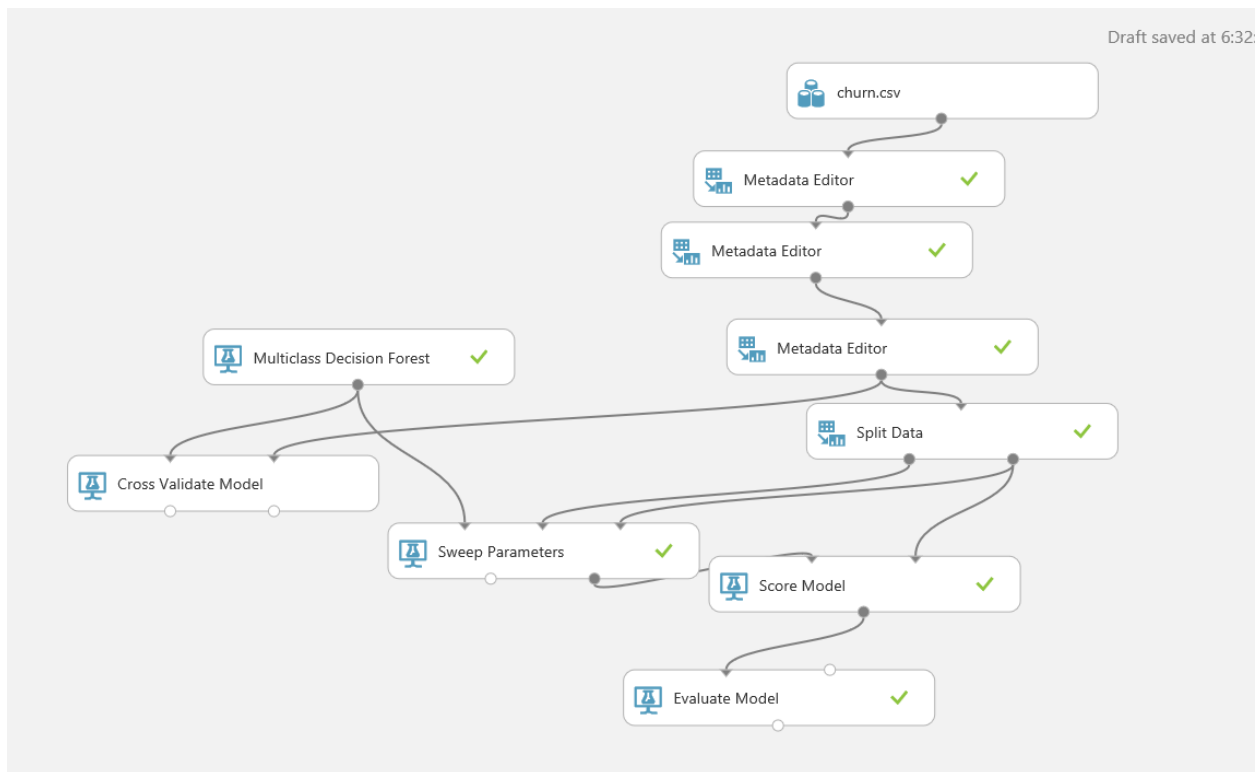
22. Click on the circle of ‘Evaluate Model’ and visualize it. You can see evaluation made just like below.

Overall accuracy	0.762873
Average accuracy	0.841915
Micro-averaged precision	0.762873
Macro-averaged precision	0.769825
Micro-averaged recall	0.762873
Macro-averaged recall	0.824488

◀ Confusion Matrix



23. Search for ‘**Cross validation model**’ and drag it onto the canvas. Make connections like screenshot below.



24. Click on the Cross Validation Module and click ‘Launch column Selector’. Just select **Churned**.

Notes: Cross validation is a way to avoid either underfitting and overfitting. It can give more reliable evaluations than split method to do the training.

25. Run the experiment again.

26. Right click on the right circle of the cross validation module and choose visualization.

Fold Number	Number of examples in fold
0	147
1	148
2	148

Notes: Azure makes 10-fold cross validation, which is why fold Numbers arrange from 0 to 9. It split randomly the data into 10 folds and train every 9 folds and test on the extract one fold.

The evaluation function here automatically lists all metrics for every class. Suppose here we just need to focus on precision.

Average Log Loss for Class "Current"	Precision for Class "Current"	Recall for Class "Current"	Average Log Loss for Class "InVol"	Precision for Class "InVol"	Recall for Class "InVol"	Average Log Loss for Class "Vol"	Precision for Class "Vol"	Recall for Class "Vol"
0.339226	0.888889	0.808989	0.360658	0.666667	0.727273	0.9931	0.703704	0.808511
1.168862	0.786667	0.766234	0.159416	0.777778	1	1.54109	0.709091	0.684211
0.31785	0.846154	0.846154	0.116698	0.761905	1	0.520942	0.795918	0.722222

27. Search for ‘**Project Columns**’ and drag it onto the canvas. Make connections with Cross Validation Module. Click on ‘Launch column selector’ and choose following columns as shown in screenshot.

×

Select columns

BY NAME

WITH RULES

AVAILABLE COLUMNS

All Types

search columns

Number of examples in fold

Model

Average Log Loss for Class "Current"

Recall for Class "Current"

Average Log Loss for Class "InVol"

Recall for Class "InVol"

Average Log Loss for Class "Vol"

Recall for Class "Vol"

8 columns available

SELECTED COLUMNS

All Types

search columns

Precision for Class "Current"

Precision for Class "InVol"

Precision for Class "Vol"




Fold Number

4 columns selected

✓

18. Run the experiment again.

19. Visualizing ‘Project Columns’ by right clicking the circle.

Fold Number	Precision for Class "Current"	Precision for Class "InVol"	Precision for Class "Vol"
			
0	0.888889	0.666667	0.703704
1	0.786667	0.777778	0.709091
2	0.846154	0.761905	0.795918
3	0.868421	0.85	0.673077
4	0.882353	0.916667	0.701493
5	0.919355	0.666667	0.726027
6	0.879121	0.727273	0.76087
7	0.873239	0.913043	0.666667
8	0.87013	0.866667	0.642857
9	0.7625	0.8	0.641509
Mean	0.857683	0.794667	0.702121
Standard Deviation	0.047835	0.091797	0.04962

Notes: ‘Mean’ means the average model precision of each fold. The Standard Deviation shows how variant our built mode performs on each fold. High standard deviation can be translated into overfitting or underfitting.

Here, the standard deviation is acceptable compared to mean value. We can say that our model doesn’t suffer serious overfitting or underfitting problems.

Below is the final view of your Azure Model:

