Machine Learning

Microsoft

Azure Machine Learning Studio Lab

- Visualization

Objectives

- Understand the matplotlib library and it's use
- Integrate and view plots using matplotlib library within the Azure Machine Learning Studio
- Develop a correlation matrix for a given dataset and view it using matplotlib

The Problem Domain

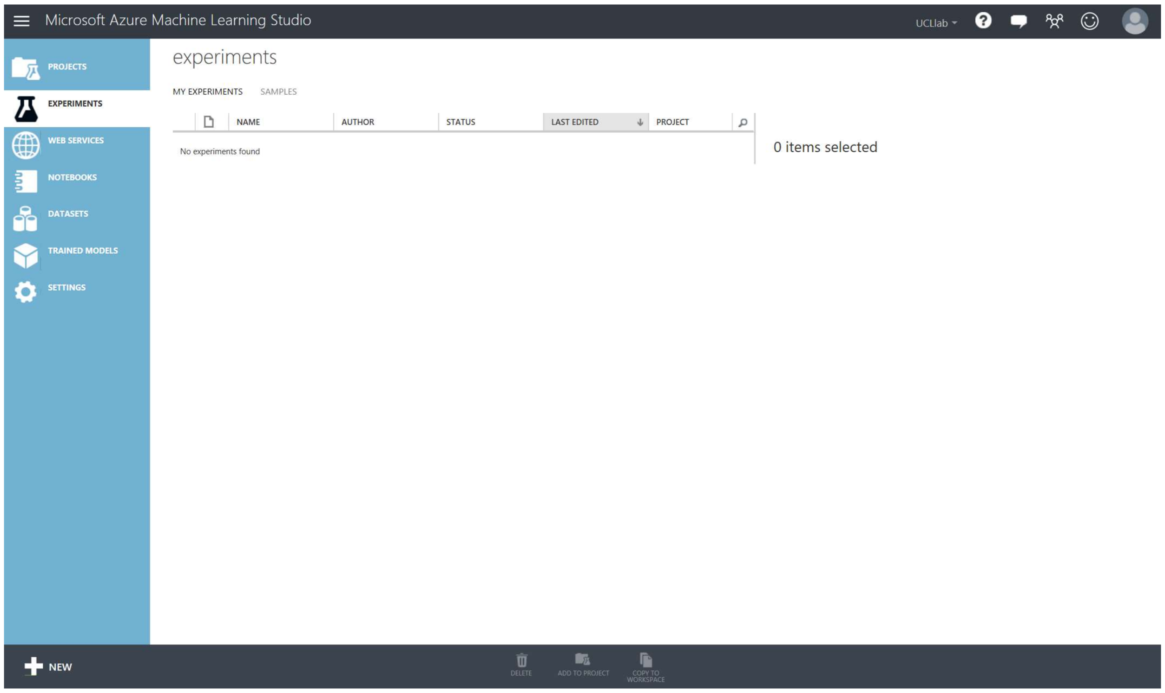
In this lab, we will use a well-known dataset related to breast cancer from the field of Oncology and has repeatedly appeared in the machine learning literature. The dataset is described by 9 attributes, some of which are linear and some are nominal. The dataset is of classification in nature capturing if recurrence-event occurred or not (i.e. the class if 0/1 indicating no recurrence and 1 for recurrence). The goal of your task is to generate a correlation matrix between the attributes visually to get an insight into the variables that are correlated between each other and with the class.

The attribute columns in the dataset include:

- age
- menopause
- tumor-size
- inv-nodes
- node-caps
- deg-malig
- breast
- breast-quad
- irradiat

Enter Workspace

Set up an account at http://studio.azureml.net if you have not. 'Sign In' to enter the Azure Machine Learning Studio as shown below and create a new experiment.

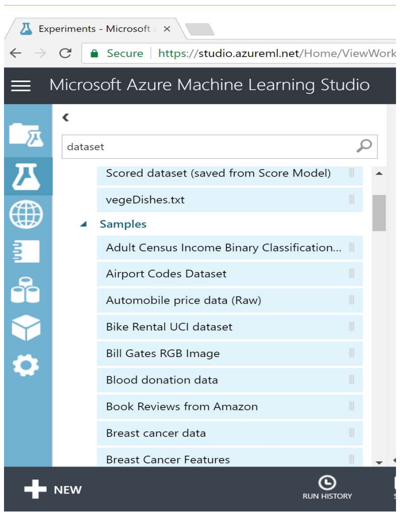


No experiments found

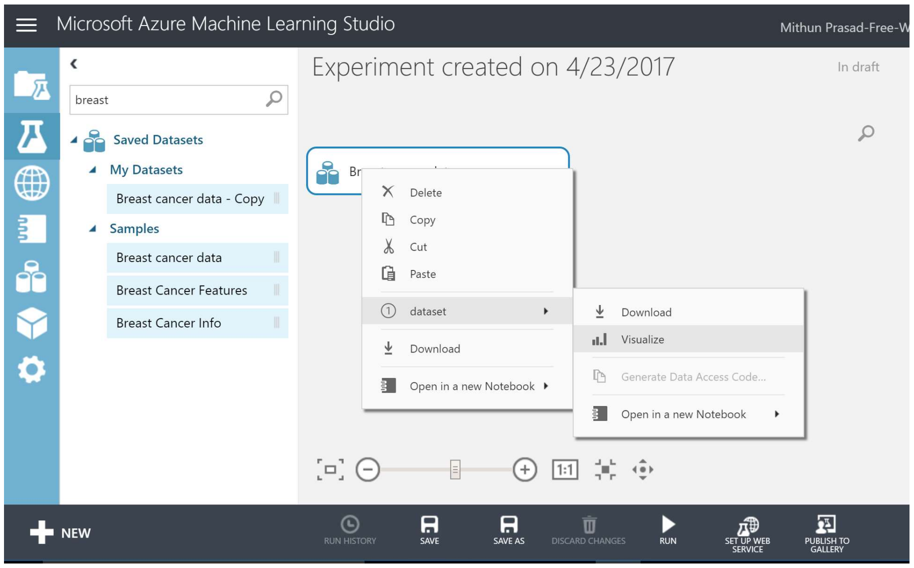
0 items selected

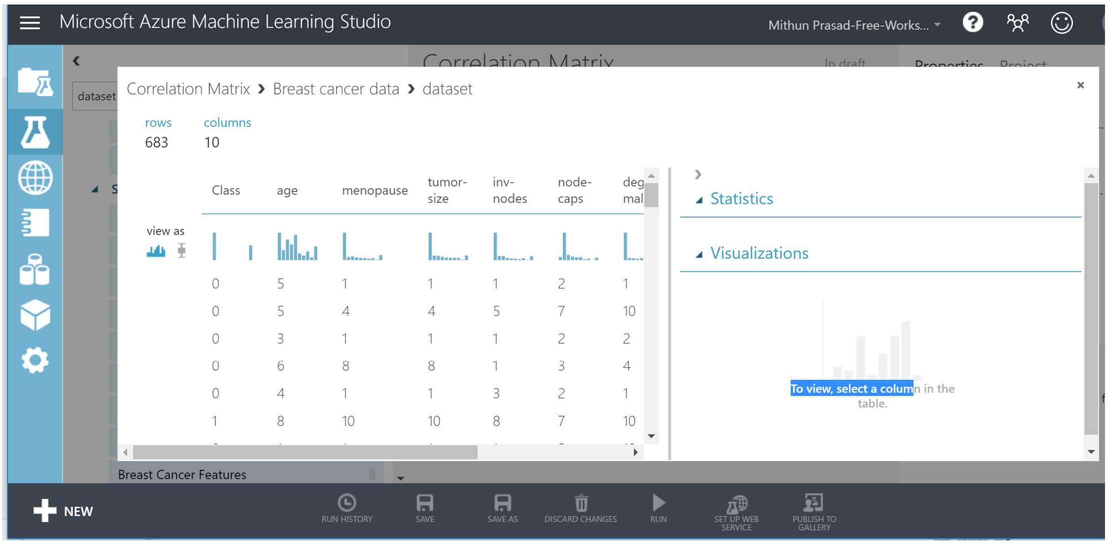
Getting Started

In studio, load the 'breast cancer data' dataset from the samples as shown below. Drag the sample dataset to the experiment area.



Visualize the dataset by right clicking on the dataset (in the experiment space) and choosing visualize as shown below.

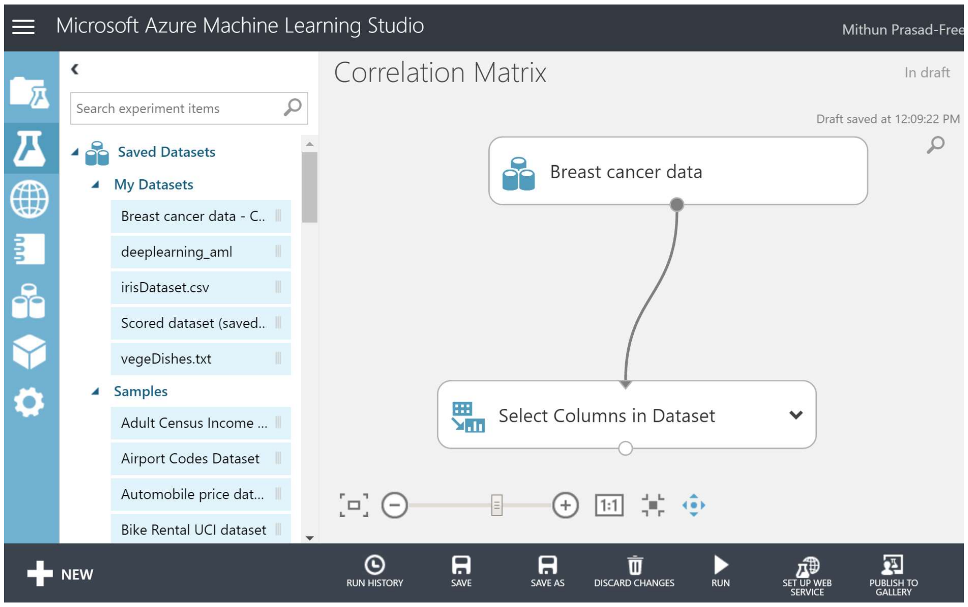


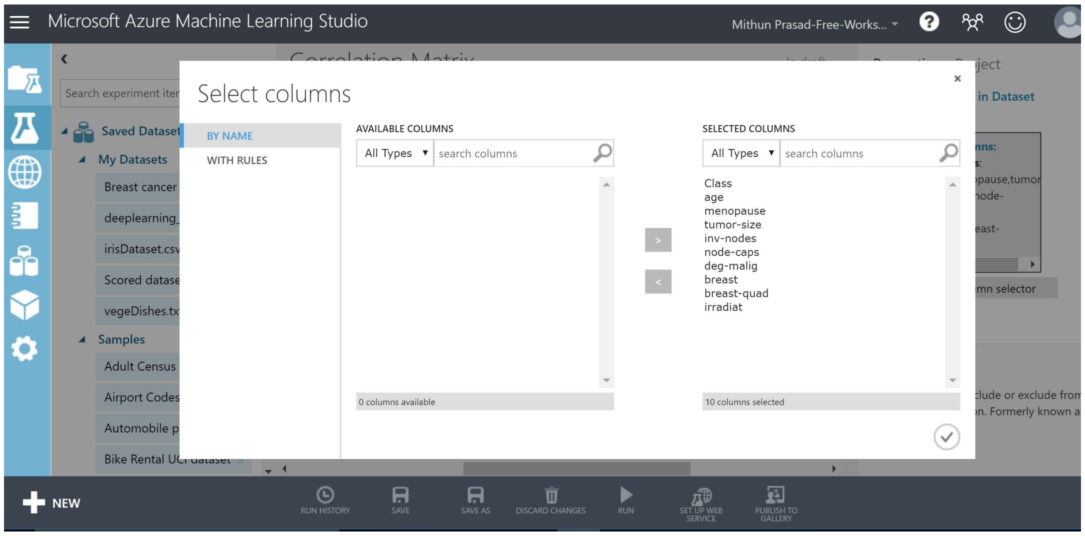

Microsoft Azure Machine Learning Studio
Mithun Prasad-Free-Works...
Correlation Matrix › Breast cancer data › dataset
Statistics
Visualizations
To view, select a column in the table.
Breast Cancer Features
NEW
RUN HISTORY
SAVE
SAVE AS
DISCARD CHANGES
RUN
SET UP WEB SERVICE
PUBLISH TO GALLERY

rows	columns
683	10

Class	age	menopause	tumor-size	inv-nodes	node-caps	deg-mal
0	5	1	1	1	2	1
0	5	4	4	5	7	10
0	3	1	1	1	2	2
0	6	8	8	1	3	4
0	4	1	1	3	2	1
1	8	10	10	8	7	10

Introduce a "Select Columns in Dataset" module in the experiment space and connect it from the dataset. Ensure that the following columns are all added. Note that the class variable also is added as we want to find out the correlation between the attributes and the class.





Select columns

BY NAME
WITH RULES

AVAILABLE COLUMNS
All Types
search columns
0 columns available

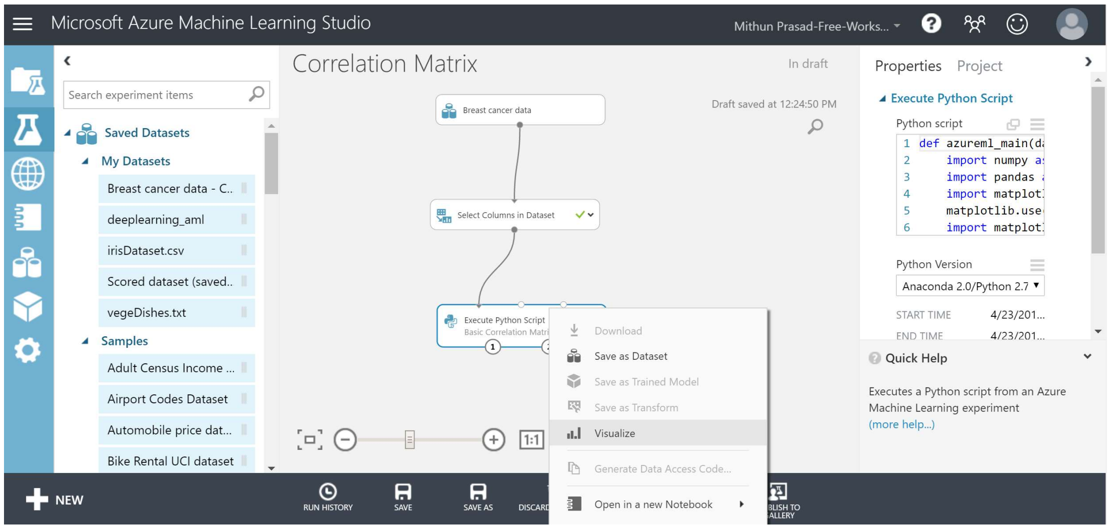
SELECTED COLUMNS
All Types
search columns

Class
age
menopause
tumor-size
inv-nodes
node-caps
deg-malig
breast
breast-quad
irradiat

10 columns selected

Visualization with Python

Connect “Execute Python Script” to the “Select Columns in Dataset module” and insert the below python code in the script box.



Python script

```
def azureml_main(d
    import numpy a
    import pandas
    import matplot
    matplotlib.use
    import matplot
```

Python Version: Anaconda 2.0/Python 2.7

START TIME 4/23/201...

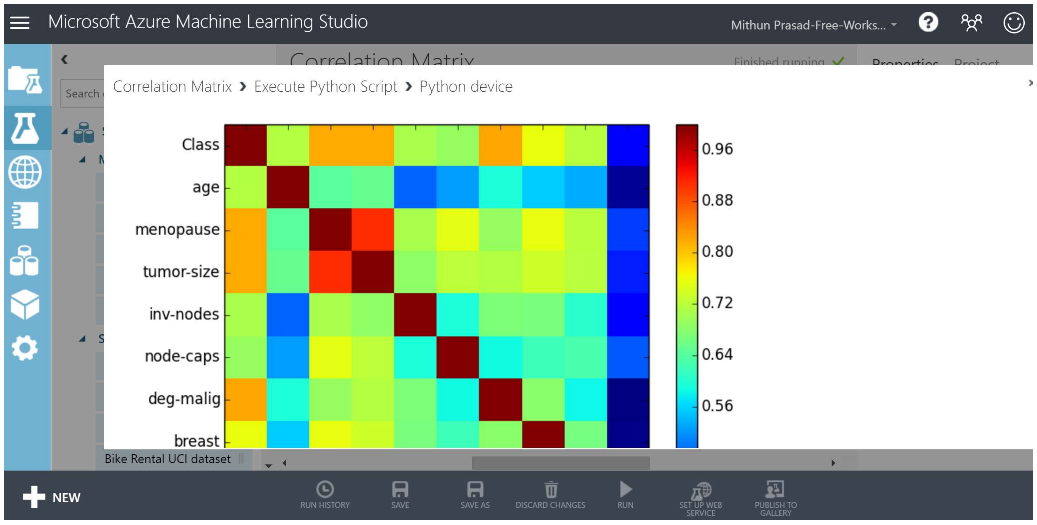
END TIME 4/23/201...

Executes a Python script from an Azure Machine Learning experiment (more help...)

```
def azureml_main(dataframe1 = None):

    import numpy as np
    import pandas as pd
    import matplotlib
    matplotlib.use("agg")
    import matplotlib.pyplot as plt
    cm=dataframe1.corr()

    fig=plt.figure()
    plt.imshow(cm,interpolation='nearest')
    plt.xticks(list(range(0,len(cm.columns))),list(cm.columns.values), rotation=45)
    plt.yticks(list(range(0,len(cm.columns))),list(cm.columns.values))

    plt.colorbar()

    fig.savefig("CorrelationPlot.png")
    return pd.DataFrame(cm)
```

Refer to matplotlib details in Appendix A for background and introduction.

In order to view the correlation heatmap visualization, right-click on "Execute Python Script" module and choose Visualize. You will find a visualization matrix heatmap as illustrated:



Notice the diagonal capturing brown boxes or a correlation of 1.

1. Which variables are strongly correlated with the class?
2. Which variables are least correlated with each other?

Conclusion

This lab introduced you to the well-known matplotlib library to develop rich graphs and visualizations. In addition, the correlation matrix can also be visualized to identify variables that are strongly correlated with the class and least correlated between other features.

Appendix A: matplotlib

What module will we use for plotting data in Python?

- MatPlotLib is the single most used Python package for 2D-graphics
- It provides both a very quick way to visualize data from Python and publication-quality figures in many formats

References

- matplotlib.org
- It is included in Anaconda

Basic concepts of a MatPlotLib plotting script

A MatPlotLib plotting script will be:

1. Import the required modules (numpy as np and matplotlib.pyplot as plt)

2. Load or generate some data

3. Customize the appearance of your figure (optional)

4. Generate a figure (plot(), bar(), pie() etc.)

5. Display it or save it in a file (many formats: PNG, PDF, SVG etc)

Backend

Matplotlib targets many different use cases and output formats. Matplotlib can be used interactively from the python shell and have plotting windows pop up when they type commands.

Matplotlib can also be embedded into graphical user interfaces like wxpython or pygtk to build rich applications. Matplotlib can also be used in batch scripts to generate postscript images from some numerical simulations.

To support all of these use cases, matplotlib can target different outputs, and each of these capabilities is called a backend; the “frontend” is the user facing code, i.e., the plotting code, whereas the “backend” does all the hard work behind-the-scenes to make the figure. There are two types of backends: user interface backends (for use in pygtk, wxpython, tkinter, qt4, or macosx;

also referred to as “interactive backends”) and hardcopy backends to make image files (PNG, SVG, PDF, PS; also referred to as “non-interactive backends”).

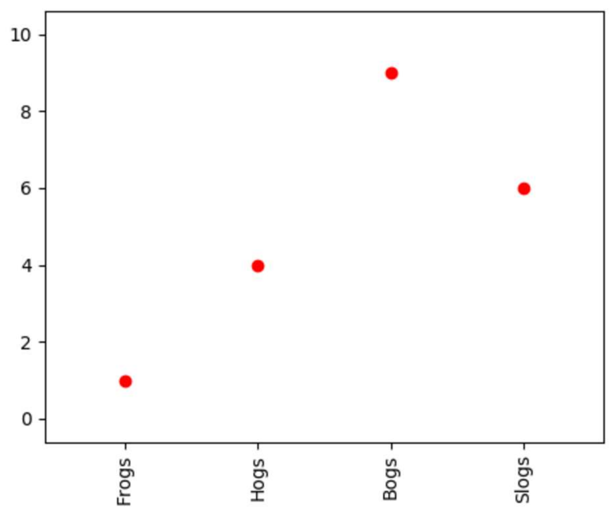
A summary of the matplotlib renderers are listed below and they can be configured by calling the `use()` command:

Renderer	Filetypes	Description
AGG	png	raster graphics – high quality images using the Anti-Grain Geometry engine
PS	ps eps	vector graphics – Postscript output
PDF	pdf	vector graphics – Portable Document Format
SVG	svg	vector graphics – Scalable Vector Graphics
Cairo	png ps pdf svg ...	vector graphics – Cairo graphics
GDK	png jpg tiff ...	raster graphics – the Gimp Drawing Kit Deprecated in 2.0

xticks and yticks:

xticks is the interval for x axis ticks or measurement and yticks is the interval for y axis ticks or measurement. See below example of how generating a simple graph using xticks and yticks.

```
import matplotlib.pyplot as plt


x = [1, 2, 3, 4]
y = [1, 4, 9, 6]
labels = ['Frogs', 'Hogs', 'Bogs', 'Slogs']

plt.plot(x, y, 'ro')
# You can specify a rotation for the tick labels in degrees or with
keywords.
plt.xticks(x, labels, rotation='vertical')
# Pad margins so that markers don't get clipped by the axes
plt.margins(0.2)
# Tweak spacing to prevent clipping of tick-labels
plt.subplots_adjust(bottom=0.15)
plt.show()
```


10
8
6
4
2
0
Frogs
Hogs
Bogs
Slogs