中原大學 雲端計算平台實務

12/17-作業報告

Microsoft Azure AI Fundamentals: Explore visual tools for machine learning

資訊四乙 10727211 林彥輝

授課教師：鍾武君 教授

中華民國一一〇年十二月

1. Learning Path Intro

Microsoft Azure AI Fundamentals: Explore visual tools for machine learning

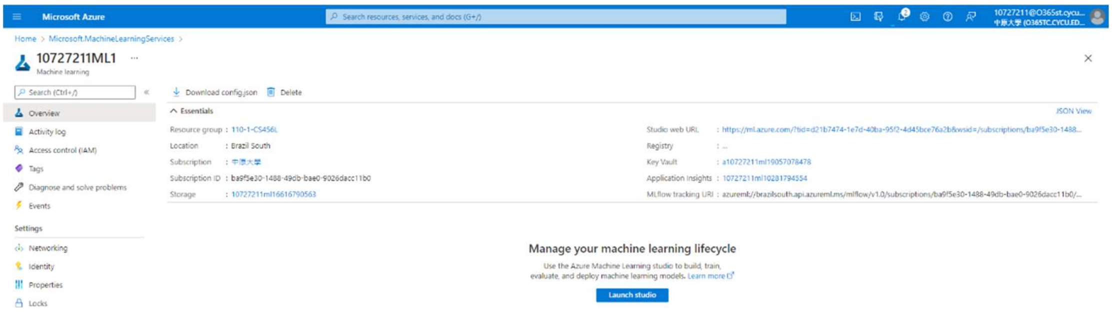
https://docs.microsoft.com/en-us/learn/paths/create-no-code-predictive-models-azure-machine-learning/

Summary Homework Assignment

Model 1: Use automated machine learning in Azure

Machine Learning

1. Create a Azure Machine Learning workspace



Resource group : 110-1-CS456L

Location : Brazil South

Subscription : 中原大學

Subscription ID : ba9f5e30-1488-49db-bae0-9026dacc11b0

Storage : 10727211ml16616790563

Studio web URL : https://ml.azure.com/?tid=d21b7474-1e7d-40ba-95f2-4d45bce76a2b&wsid=/subscriptions/ba9f5e30-1488...

Registry : ...

Key Vault : a10727211ml19057078478

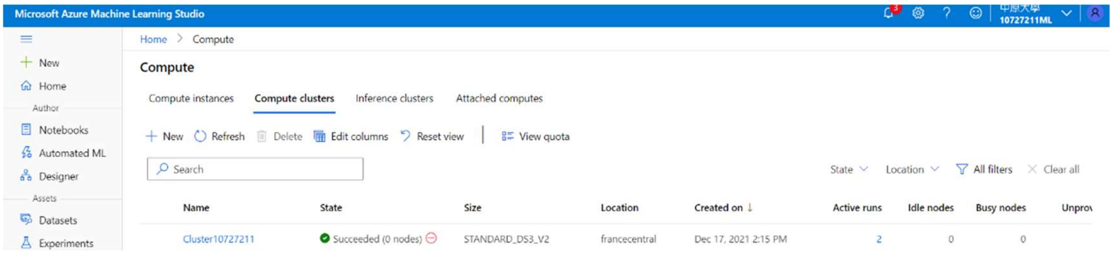
Application Insights : 10727211ml10281794554

MLflow tracking URI : azureml://brazilsouth.api.azureml.ms/mlflow/v1.0/subscriptions/ba9f5e30-1488-49db-bae0-9026dacc11b0/...

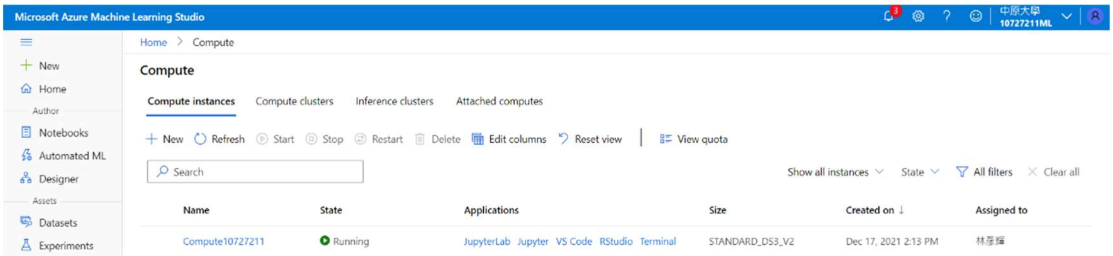
Manage your machine learning lifecycle

Use the Azure Machine Learning studio to build, train, evaluate, and deploy machine learning models. Learn more

2. Create compute resource

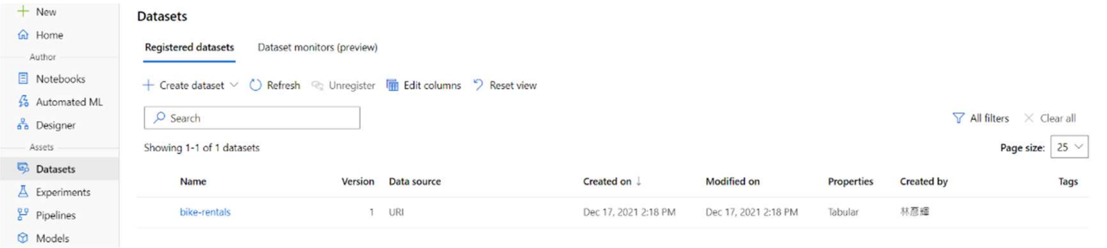


Name	State	Size	Location	Created on ↓	Active runs	Idle nodes	Busy nodes	Unprov
Cluster10727211	Succeeded (0 nodes)	STANDARD_DS3_V2	francecentral	Dec 17, 2021 2:15 PM	2	0	0	



Name	State	Applications	Size	Created on ↓	Assigned to
Compute10727211	Running	JupyterLab Jupyter VS Code RStudio Terminal	STANDARD_DS3_V2	Dec 17, 2021 2:13 PM	林彥輝

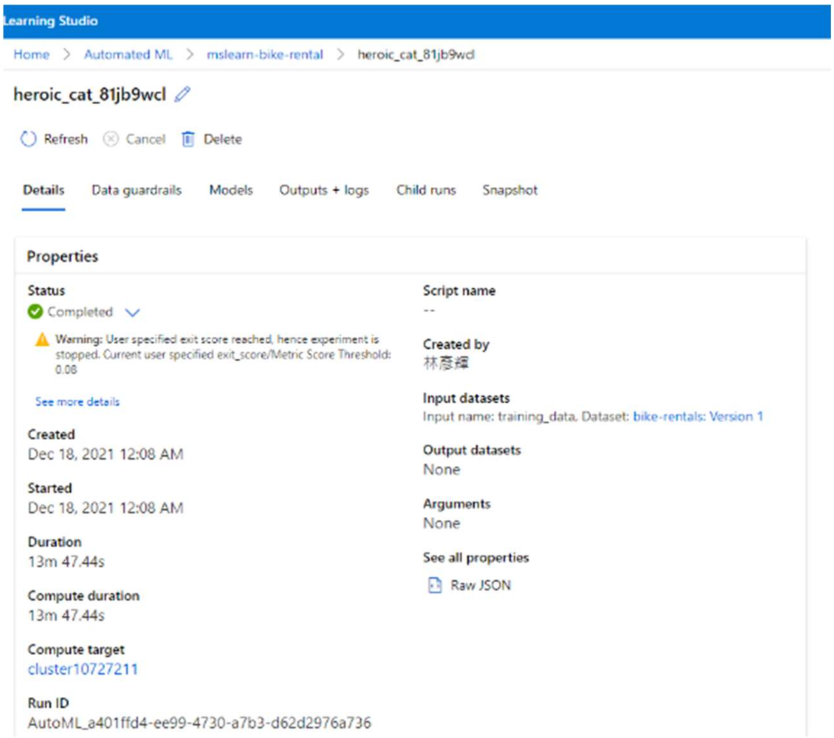
3. Create a dataset



Showing 1-1 of 1 datasets

Name	Version	Data source	Created on ↓	Modified on	Properties	Created by	Tags
bike-rentals	1	URI	Dec 17, 2021 2:18 PM	Dec 17, 2021 2:18 PM	Tabular	林彥輝	

4. Run an automated machine learning experiment



heroic_cat_81jb9wcl

Properties

Status
Completed

Warning: User specified exit score reached, hence experiment is stopped. Current user specified exit_score/Metric Score Threshold: 0.08

See more details

Created
Dec 18, 2021 12:08 AM

Started
Dec 18, 2021 12:08 AM

Duration
13m 47.44s

Compute duration
13m 47.44s

Compute target
cluster10727211

Run ID
AutoML_a401ffd4-ee99-4730-a7b3-d62d2976a736

Script name
--

Created by
林彥輝

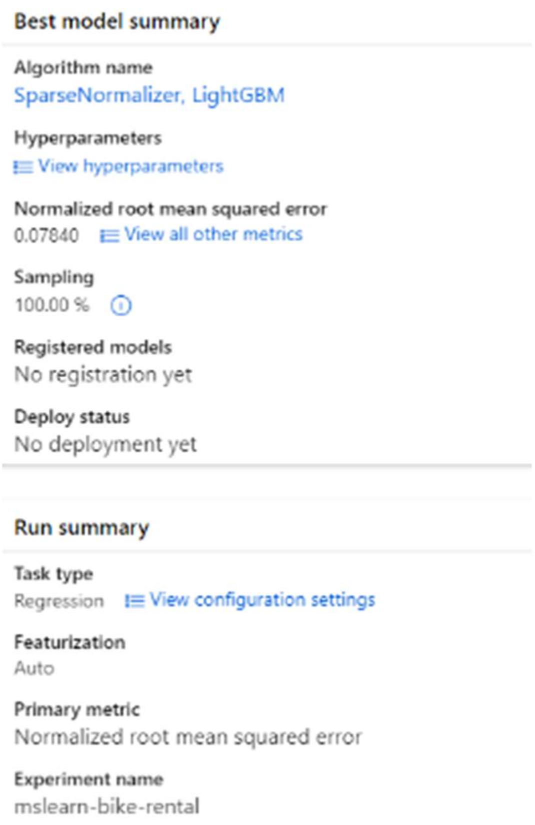
Input datasets
Input name: training_data, Dataset: bike-rentals: Version 1

Output datasets
None

Arguments
None

See all properties
Raw JSON

5. Review the best model



Best model summary

Algorithm name
SparseNormalizer, LightGBM

Hyperparameters

Normalized root mean squared error
0.07840

Sampling
100.00 %

Registered models
No registration yet

Deploy status
No deployment yet

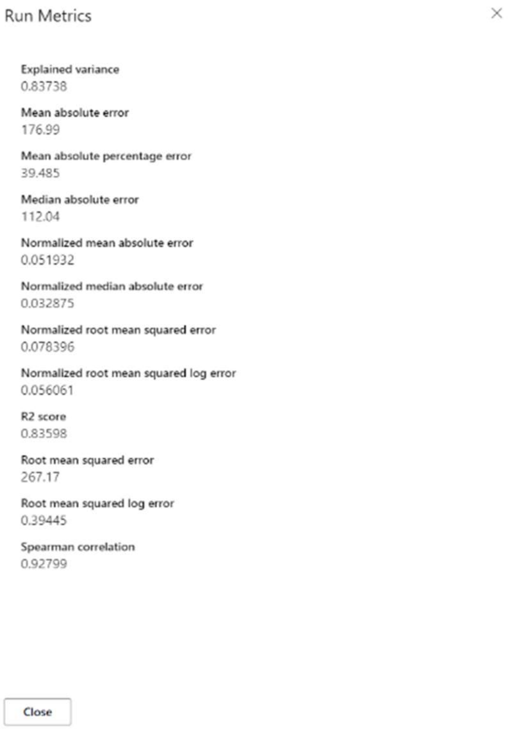
Run summary

Task type
Regression

Featurization
Auto

Primary metric
Normalized root mean squared error

Experiment name
mslearn-bike-rental



Run Metrics

Explained variance
0.83738

Mean absolute error
176.99

Mean absolute percentage error
39.485

Median absolute error
112.04

Normalized mean absolute error
0.051932

Normalized median absolute error
0.032875

Normalized root mean squared error
0.078396

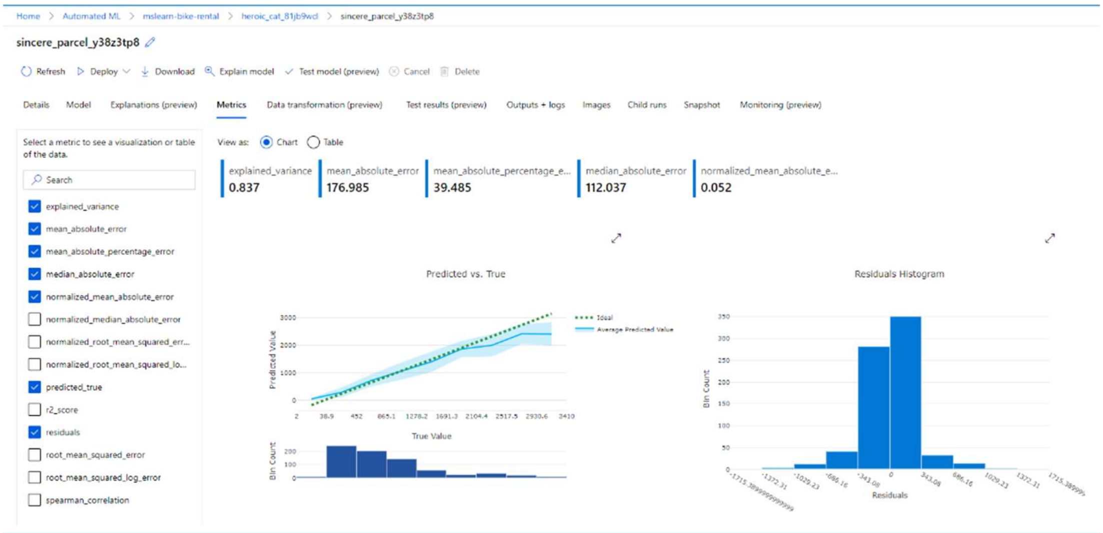
Normalized root mean squared log error
0.056061

R2 score
0.83598

Root mean squared error
267.17

Root mean squared log error
0.39445

Spearman correlation
0.92799



Home > Automated ML > mslearn-bike-rental > heroic_cat_81jb9wcl > sincere_parcel_y38z3tp8

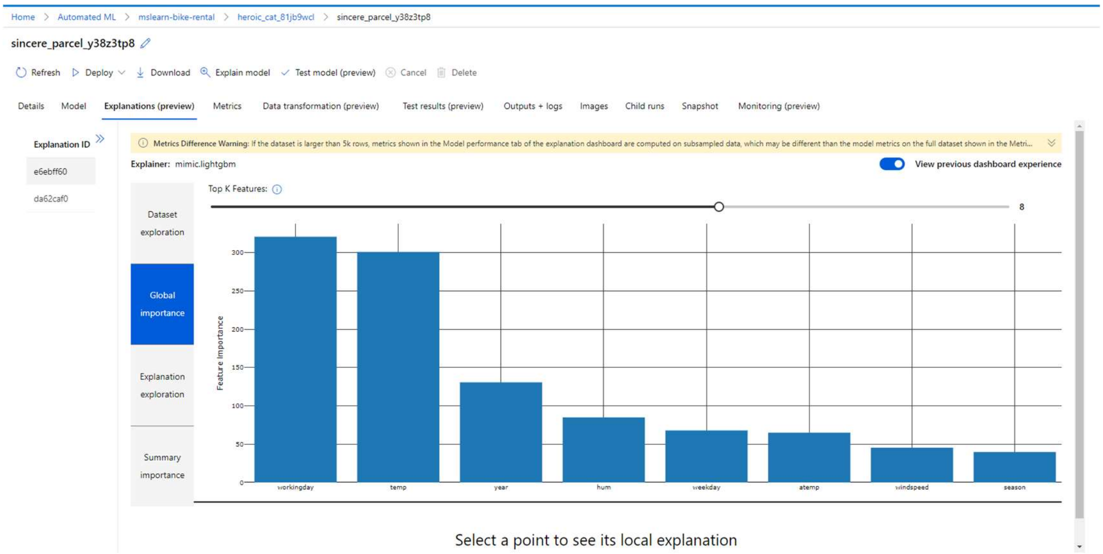
sincere_parcel_y38z3tp8

Select a metric to see a visualization or table of the data.

- ☑ explained_variance
- ☑ mean_absolute_error
- ☑ mean_absolute_percentage_error
- ☑ median_absolute_error
- ☑ normalized_mean_absolute_error
- □ normalized_median_absolute_error
- □ normalized_root_mean_squared_err...
- □ normalized_root_mean_squared_lo...
- ☑ predicted_true
- □ r2_score
- ☑ residuals
- □ root_mean_squared_error
- □ root_mean_squared_log_error
- □ spearman_correlation

View as: ● Chart ○ Table

explained_variance	mean_absolute_error	mean_absolute_percentage_e...	median_absolute_error	normalized_mean_absolute_e...
0.837	176.985	39.485	112.037	0.052



Home > Automated ML > mslearn-bike-rental > heroic_cat_81jb9wcl > sincere_parcel_y38z3tp8

sincere_parcel_y38z3tp8

Metrics Difference Warning: If the dataset is larger than 5k rows, metrics shown in the Model performance tab of the explanation dashboard are computed on subsampled data, which may be different than the model metrics on the full dataset shown in the Metri...

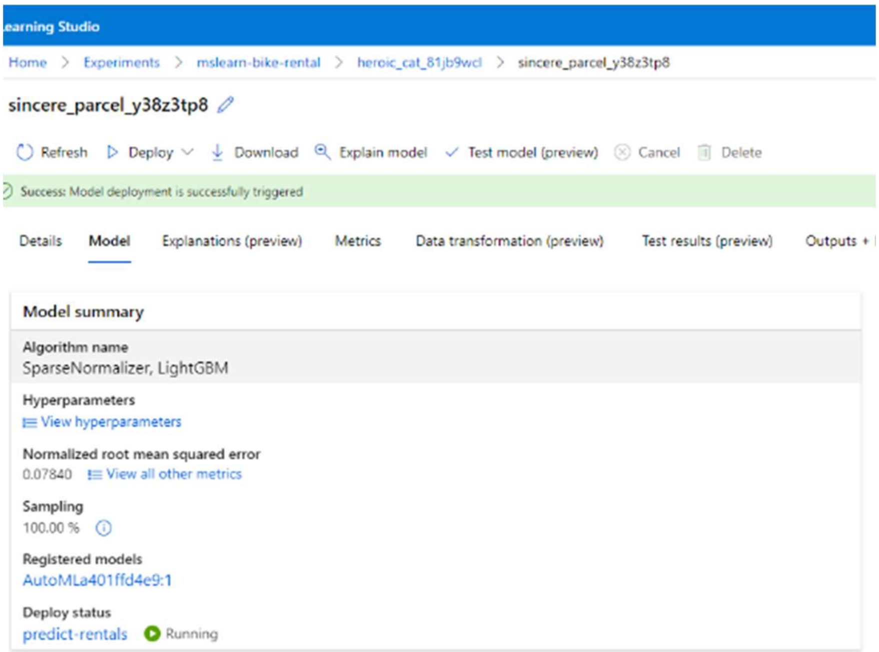
Explainer: mimic.lightgbm

View previous dashboard experience

Top K Features: 8

Select a point to see its local explanation

6. Deploy a model as a service



earning Studio

Home > Experiments > mslearn-bike-rental > heroic_cat_81jb9wcl > sincere_parcel_y38z3tp8

sincere_parcel_y38z3tp8

Success: Model deployment is successfully triggered

Model summary

Algorithm name
SparseNormalizer, LightGBM

Hyperparameters
View hyperparameters

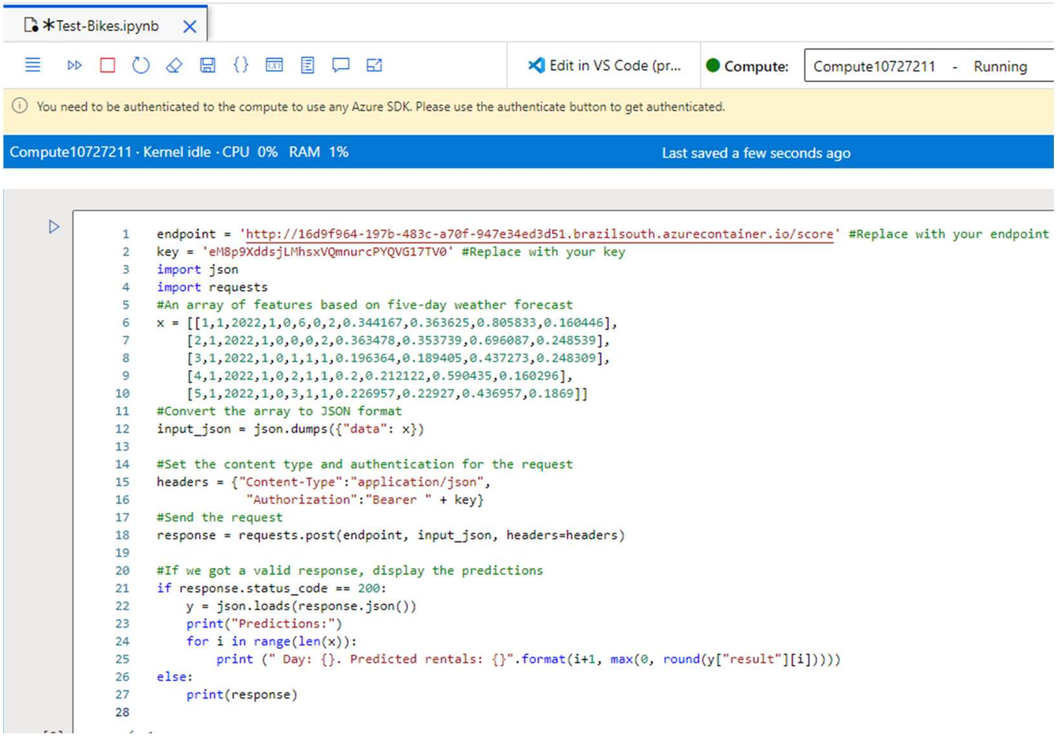
Normalized root mean squared error
0.07840 View all other metrics

Sampling
100.00 %

Registered models
AutoMLa401ffd4e9:1

Deploy status
predict-rentals Running

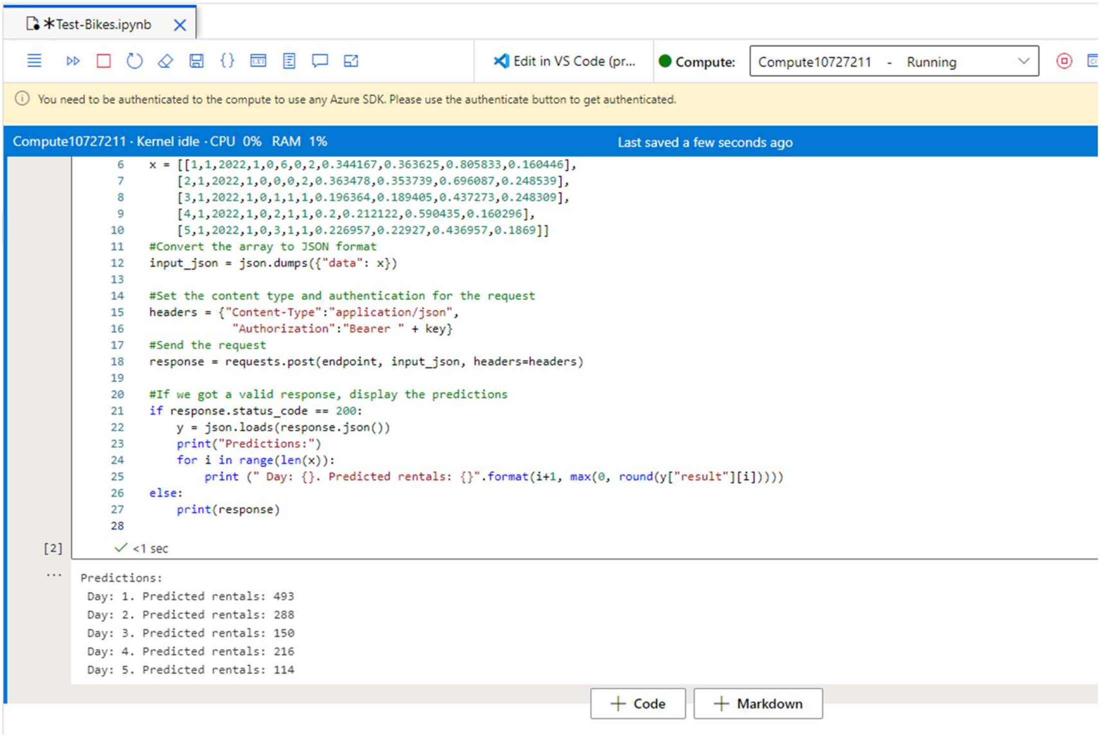
7. Test the deployed service



You need to be authenticated to the compute to use any Azure SDK. Please use the authenticate button to get authenticated.

Compute10727211 · Kernel idle · CPU 0% RAM 1%   Last saved a few seconds ago

```
endpoint = 'http://16d9f964-197b-483c-a70f-947e34ed3d51.brazilsouth.azurecontainer.io/score' #Replace with your endpoint
key = 'eM8p9XddsjLMhsxVQmnurcPYQVG17TV0' #Replace with your key
import json
import requests
#An array of features based on five-day weather forecast
x = [[1,1,2022,1,0,6,0,2,0.344167,0.363625,0.805833,0.160446],
    [2,1,2022,1,0,0,0,2,0.363478,0.353739,0.696087,0.248539],
    [3,1,2022,1,0,1,1,1,0.196364,0.189405,0.437273,0.248309],
    [4,1,2022,1,0,2,1,1,0.2,0.212122,0.590435,0.160296],
    [5,1,2022,1,0,3,1,1,0.226957,0.22927,0.436957,0.1869]]
#Convert the array to JSON format
input_json = json.dumps({"data": x})

#Set the content type and authentication for the request
headers = {"Content-Type":"application/json",
            "Authorization":"Bearer " + key}
#Send the request
response = requests.post(endpoint, input_json, headers=headers)

#If we got a valid response, display the predictions
if response.status_code == 200:
    y = json.loads(response.json())
    print("Predictions:")
    for i in range(len(x)):
        print (" Day: {}. Predicted rentals: {}".format(i+1, max(0, round(y["result"][i]))))
else:
    print(response)

```



You need to be authenticated to the compute to use any Azure SDK. Please use the authenticate button to get authenticated.

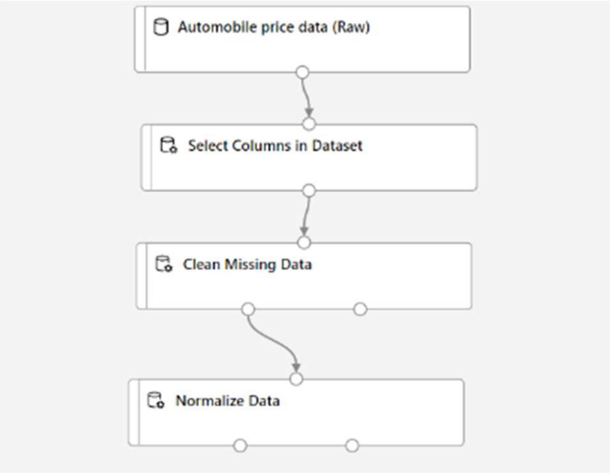
Compute10727211 · Kernel idle · CPU 0% RAM 1%   Last saved a few seconds ago

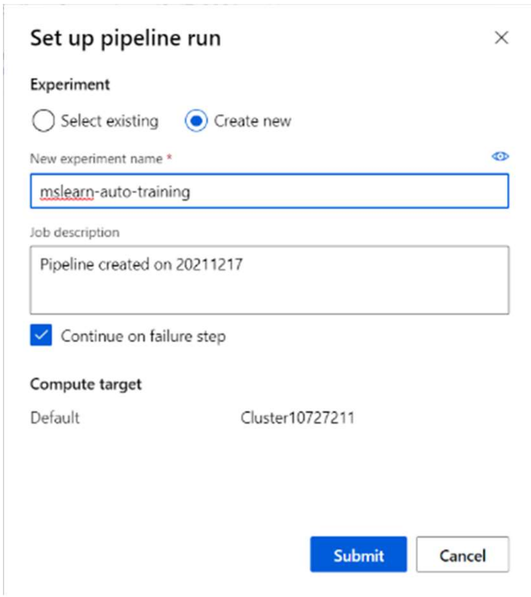
```
x = [[1,1,2022,1,0,6,0,2,0.344167,0.363625,0.805833,0.160446],
    [2,1,2022,1,0,0,0,2,0.363478,0.353739,0.696087,0.248539],
    [3,1,2022,1,0,1,1,1,0.196364,0.189405,0.437273,0.248309],
    [4,1,2022,1,0,2,1,1,0.2,0.212122,0.590435,0.160296],
    [5,1,2022,1,0,3,1,1,0.226957,0.22927,0.436957,0.1869]]
#Convert the array to JSON format
input_json = json.dumps({"data": x})

#Set the content type and authentication for the request
headers = {"Content-Type":"application/json",
            "Authorization":"Bearer " + key}
#Send the request
response = requests.post(endpoint, input_json, headers=headers)

#If we got a valid response, display the predictions
if response.status_code == 200:
    y = json.loads(response.json())
    print("Predictions:")
    for i in range(len(x)):
        print (" Day: {}. Predicted rentals: {}".format(i+1, max(0, round(y["result"][i]))))
else:
    print(response)

```

```
Predictions:
 Day: 1. Predicted rentals: 493
 Day: 2. Predicted rentals: 288
 Day: 3. Predicted rentals: 150
 Day: 4. Predicted rentals: 216
 Day: 5. Predicted rentals: 114
```

Model 2:Create a Regression model with Azure Machine Learning designer

1. Create pipeline & Add and explore dataset & Run pipeline





Set up pipeline run

Experiment

○ Select existing ● Create new

New experiment name *

mslearn-auto-training

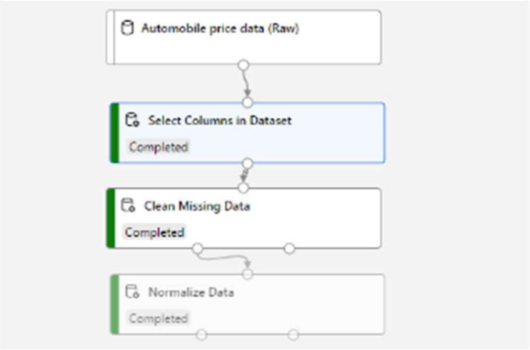
Job description

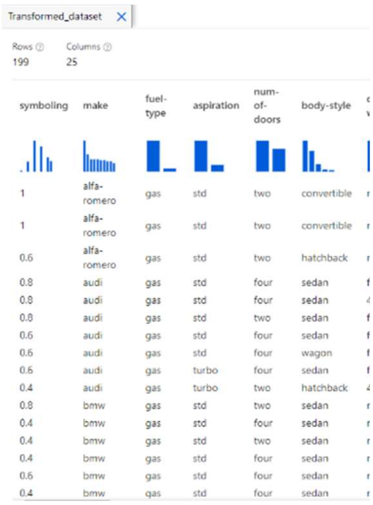
Pipeline created on 20211217

☑ Continue on failure step

Compute target

Default Cluster10727211

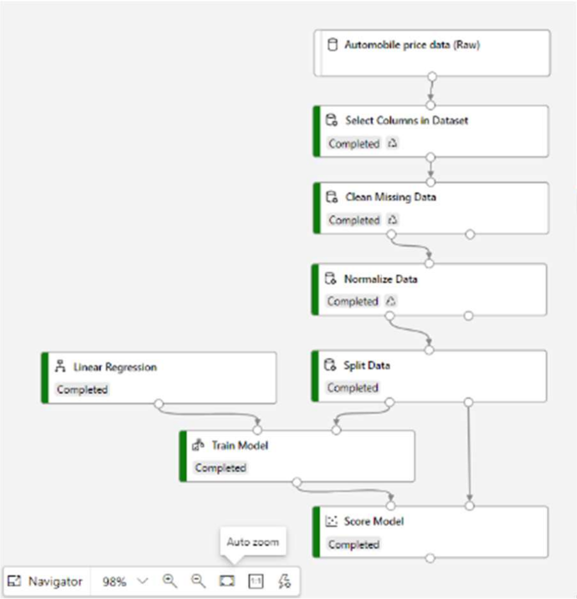


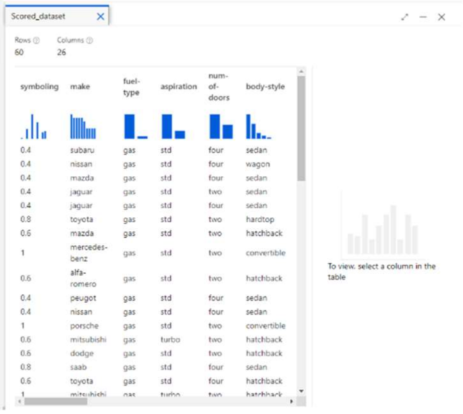


Rows 199 Columns 25

symboling	make	fuel-type	aspiration	num-of-doors	body-style
1	alfa-romero	gas	std	two	convertible
1	alfa-romero	gas	std	two	convertible
0.6	alfa-romero	gas	std	two	hatchback
0.8	audi	gas	std	four	sedan
0.8	audi	gas	std	four	sedan
0.8	audi	gas	std	two	sedan
0.6	audi	gas	std	four	sedan
0.6	audi	gas	std	four	wagon
0.6	audi	gas	turbo	four	sedan
0.4	audi	gas	turbo	two	hatchback
0.8	bmw	gas	std	two	sedan
0.4	bmw	gas	std	four	sedan
0.4	bmw	gas	std	two	sedan
0.4	bmw	gas	std	four	sedan
0.6	bmw	gas	std	four	sedan
0.4	bmw	gas	std	four	sedan

2. Create and run a training pipeline



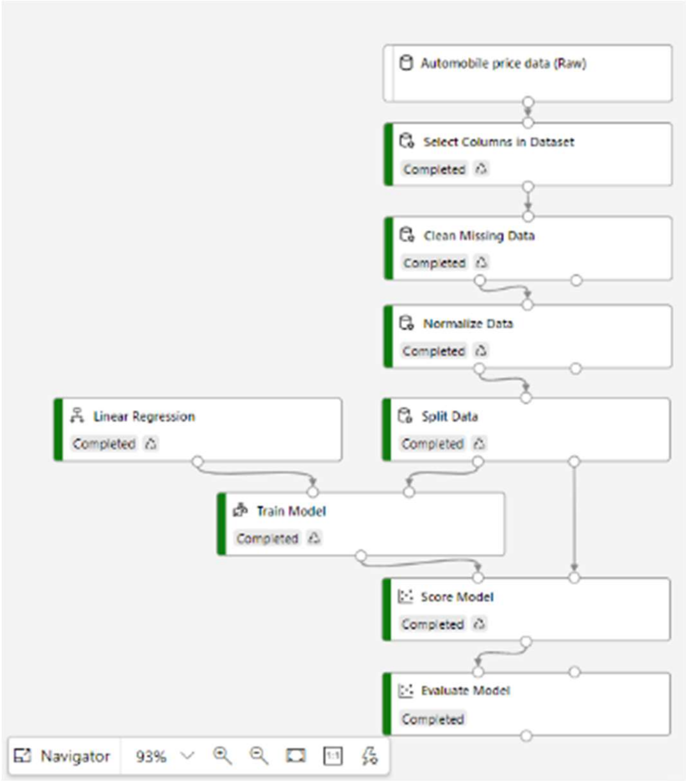

Scored_dataset

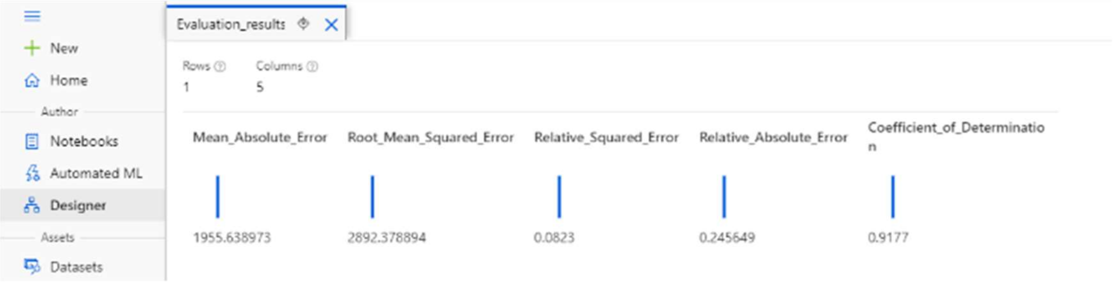
Rows 60   Columns 26

symboling	make	fuel-type	aspiration	num-of-doors	body-style
0.4	subaru	gas	std	four	sedan
0.4	nissan	gas	std	four	wagon
0.4	mazda	gas	std	four	sedan
0.4	jaguar	gas	std	two	sedan
0.4	jaguar	gas	std	four	sedan
0.8	toyota	gas	std	two	hardtop
0.6	mazda	gas	std	two	hatchback
1	mercedes-benz	gas	std	two	convertible
0.6	alfa-romero	gas	std	two	hatchback
0.4	peugot	gas	std	four	sedan
0.4	nissan	gas	std	four	sedan
1	porsche	gas	std	two	convertible
0.6	mitsubishi	gas	turbo	two	hatchback
0.6	dodge	gas	std	two	hatchback
0.8	saab	gas	std	four	sedan
0.6	toyota	gas	std	four	hatchback
1	mitsubishi	gas	turbo	two	hatchback

To view. select a column in the table

3. Evaluate a regression model



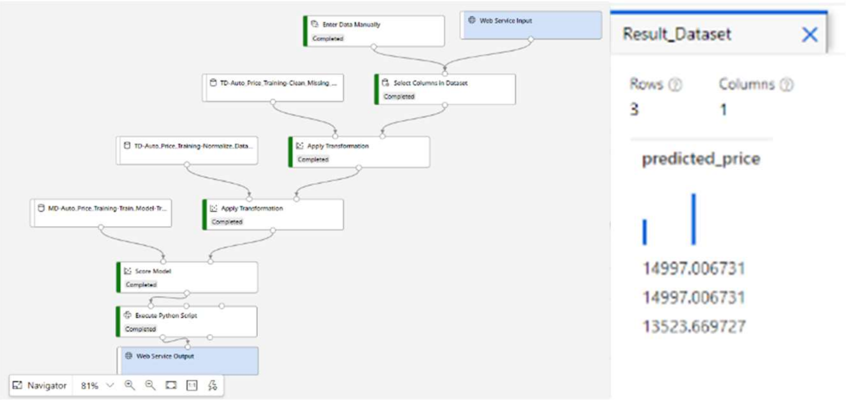


Evaluation_results

Rows 1   Columns 5

Mean_Absolute_Error	Root_Mean_Squared_Error	Relative_Squared_Error	Relative_Absolute_Error	Coefficient_of_Determination
1955.638973	2892.378894	0.0823	0.245649	0.9177

4. Create and run an inference model



Result_Dataset

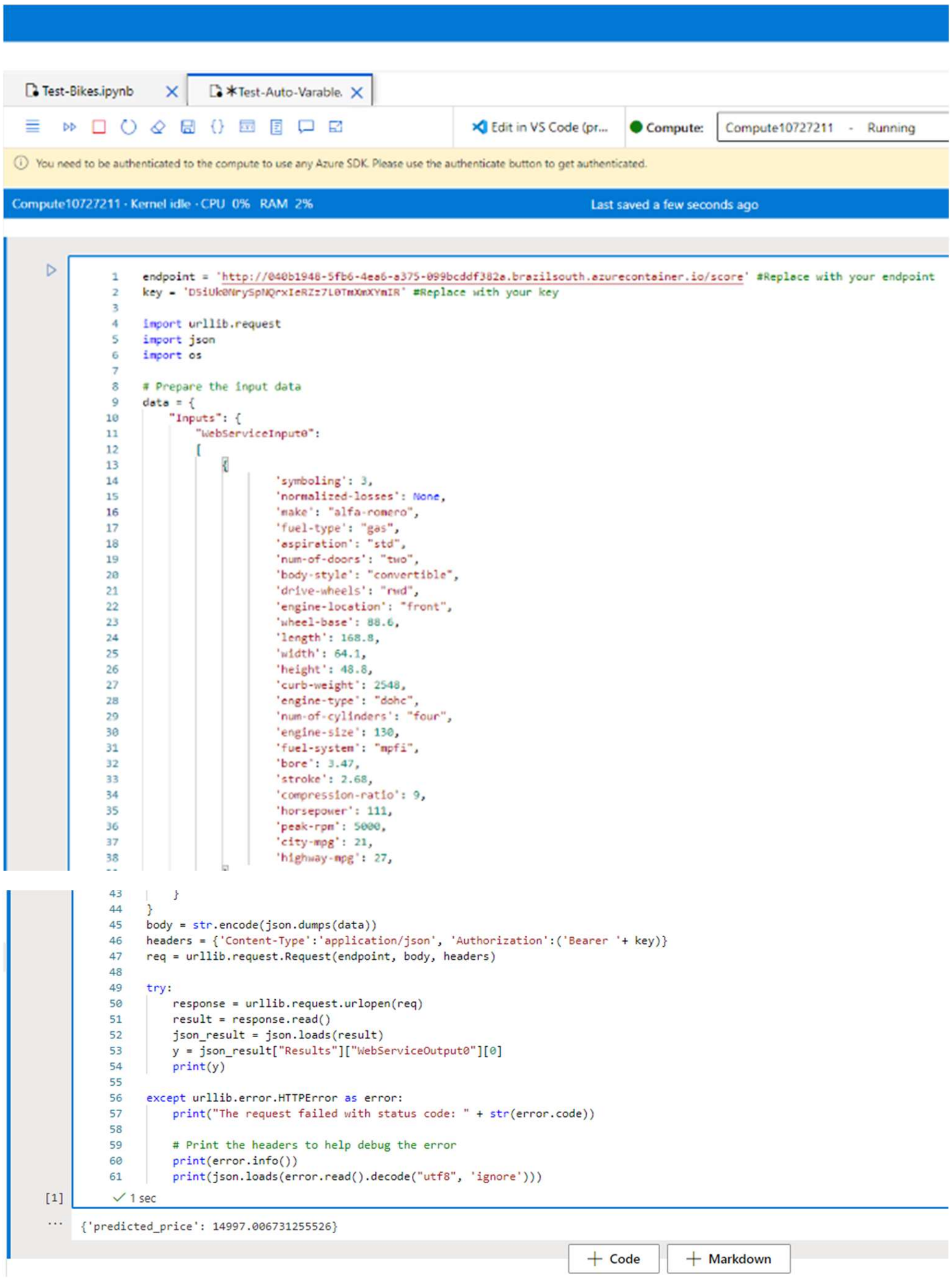
Rows 3

Columns 1

predicted_price

14997.006731
14997.006731
13523.669727

5. Deploy a predictive service & Test the service



You need to be authenticated to the compute to use any Azure SDK. Please use the authenticate button to get authenticated.

Compute10727211 · Kernel idle · CPU 0% RAM 2%

Last saved a few seconds ago

```
endpoint = 'http://040b1948-5fb6-4ea6-a375-099bcddf382a.brazilsouth.azurecontainer.io/score' #Replace with your endpoint
key = 'D5iUk0NrySpNQrxIeRZz7L0TmXmXYmIR' #Replace with your key

import urllib.request
import json
import os

# Prepare the input data
data = {
    "Inputs": {
        "WebServiceInput0":
        [
            {
                    'symboling': 3,
                    'normalized-losses': None,
                    'make': "alfa-romero",
                    'fuel-type': "gas",
                    'aspiration': "std",
                    'num-of-doors': "two",
                    'body-style': "convertible",
                    'drive-wheels': "rwd",
                    'engine-location': "front",
                    'wheel-base': 88.6,
                    'length': 168.8,
                    'width': 64.1,
                    'height': 48.8,
                    'curb-weight': 2548,
                    'engine-type': "dohc",
                    'num-of-cylinders': "four",
                    'engine-size': 130,
                    'fuel-system': "mpfi",
                    'bore': 3.47,
                    'stroke': 2.68,
                    'compression-ratio': 9,
                    'horsepower': 111,
                    'peak-rpm': 5000,
                    'city-mpg': 21,
                    'highway-mpg': 27,
```

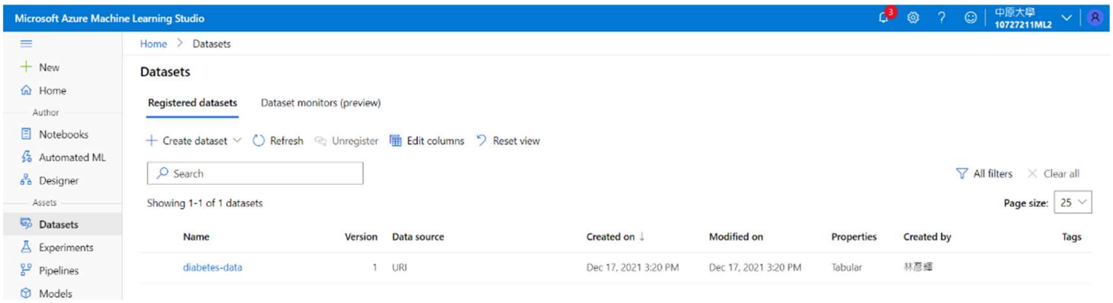
```
    }
}
body = str.encode(json.dumps(data))
headers = {'Content-Type':'application/json', 'Authorization':('Bearer '+ key)}
req = urllib.request.Request(endpoint, body, headers)

try:
    response = urllib.request.urlopen(req)
    result = response.read()
    json_result = json.loads(result)
    y = json_result["Results"]["WebServiceOutput0"][0]
    print(y)

except urllib.error.HTTPError as error:
    print("The request failed with status code: " + str(error.code))

    # Print the headers to help debug the error
    print(error.info())
    print(json.loads(error.read().decode("utf8", 'ignore')))
```

[1] ✓ 1 sec

```
{'predicted_price': 14997.006731255526}
```

Model 3:Create a classification model with Azure Machine Learning designer

1. Create a dataset

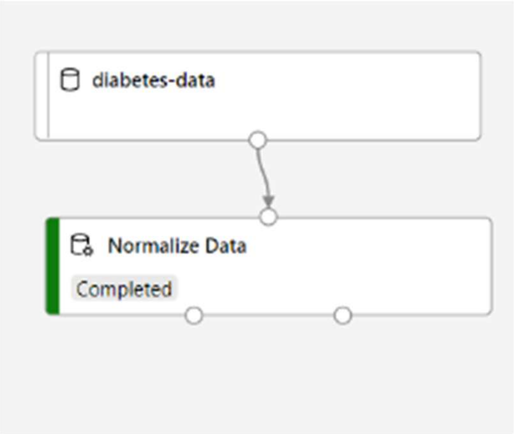


Datasets

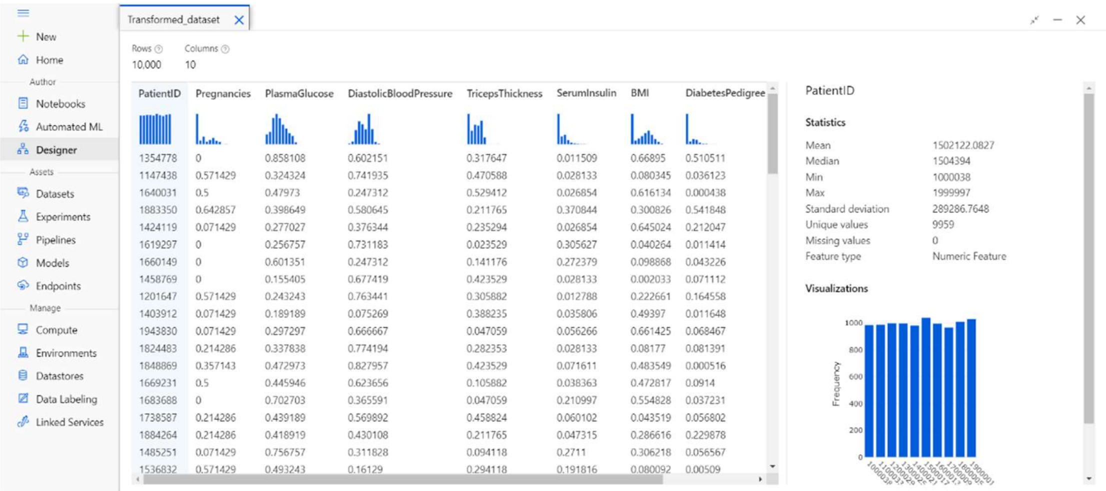
Showing 1-1 of 1 datasets

Name	Version	Data source	Created on ↓	Modified on	Properties	Created by	Tags
diabetes-data	1	URI	Dec 17, 2021 3:20 PM	Dec 17, 2021 3:20 PM	Tabular	林彥樺	

2. Create a pipeline & Add Transformations



3. Run the pipeline & View the transform data



Rows 10,000 Columns 10

PatientID	Pregnancies	PlasmaGlucose	DiastolicBloodPressure	TricepsThickness	SerumInsulin	BMI	DiabetesPedigree
1354778	0	0.858108	0.602151	0.317647	0.011509	0.66895	0.510511
1147438	0.571429	0.324324	0.741935	0.470588	0.028133	0.080345	0.036123
1640031	0.5	0.47973	0.247312	0.529412	0.026854	0.616134	0.000438
1883350	0.642857	0.398649	0.580645	0.211765	0.370844	0.300826	0.541848
1424119	0.071429	0.277027	0.376344	0.235294	0.026854	0.645024	0.212047
1619297	0	0.256757	0.731183	0.023529	0.305627	0.040264	0.011414
1660149	0	0.601351	0.247312	0.141176	0.272379	0.098868	0.043226
1458769	0	0.155405	0.677419	0.423529	0.028133	0.002033	0.071112
1201647	0.571429	0.243243	0.763441	0.305882	0.012788	0.222661	0.164558
1403912	0.071429	0.189189	0.075269	0.388235	0.035806	0.49397	0.011648
1943830	0.071429	0.297297	0.666667	0.047059	0.056266	0.661425	0.068467
1824483	0.214286	0.337838	0.774194	0.282353	0.028133	0.08177	0.081391
1848869	0.357143	0.472973	0.827957	0.423529	0.071611	0.483549	0.000516
1669231	0.5	0.445946	0.623656	0.105882	0.038363	0.472817	0.0914
1683688	0	0.702703	0.365591	0.047059	0.210997	0.554828	0.037231
1738587	0.214286	0.439189	0.569892	0.458824	0.060102	0.043519	0.056802
1884264	0.214286	0.418919	0.430108	0.211765	0.047315	0.286616	0.229878
1485251	0.071429	0.756757	0.311828	0.094118	0.2711	0.306218	0.056567
1536832	0.571429	0.493243	0.16129	0.294118	0.191816	0.080092	0.00509

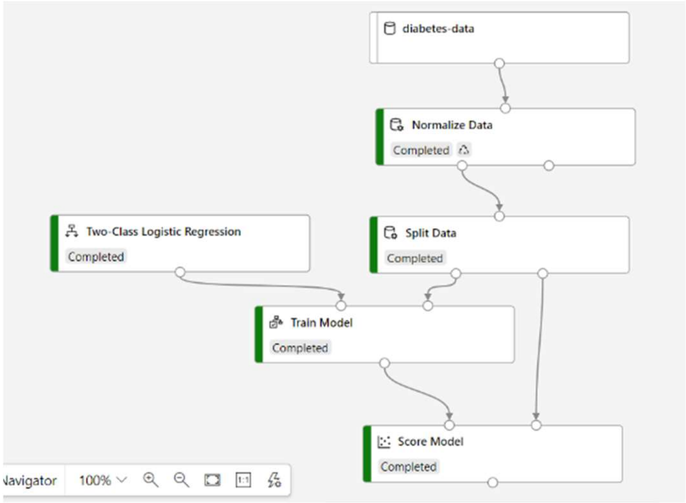
PatientID

Statistics

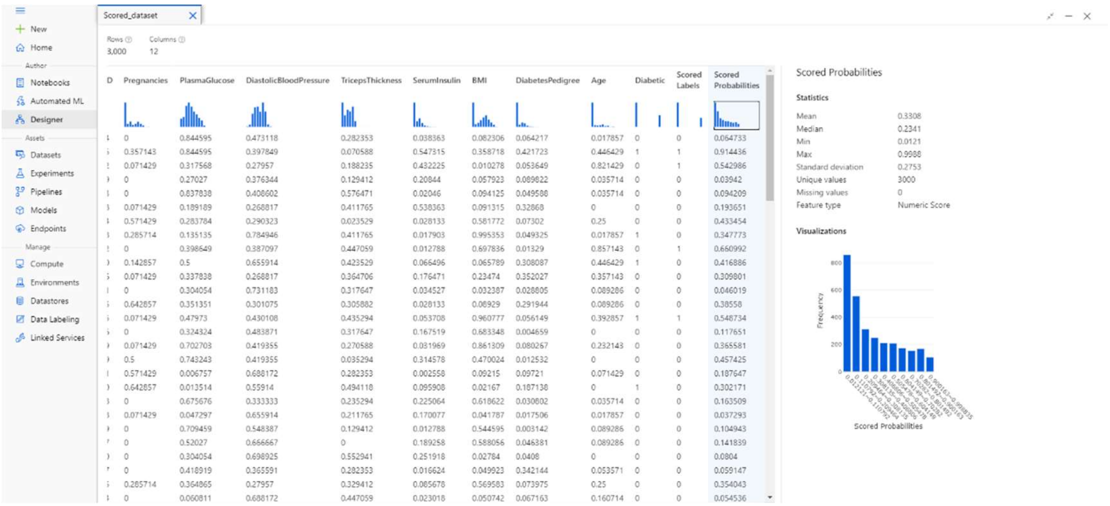
Mean	1502122.0827
Median	1504394
Min	1000038
Max	1999997
Standard deviation	289286.7648
Unique values	9959
Missing values	0
Feature type	Numeric Feature

Visualizations

4. Add training models



5. Run the training pipeline & View result



Rows 3,000 Columns 12

D	Pregnancies	PlasmaGlucose	DiastolicBloodPressure	TricepsThickness	SerumInsulin	BMI	DiabetesPedigree	Age	Diabetic	Scored Labels	Scored Probabilities
	0	0.844595	0.473118	0.282353	0.038363	0.082306	0.064217	0.017857	0	0	0.064733
	0.357143	0.844595	0.397849	0.070588	0.547315	0.358718	0.421723	0.446429	1	1	0.914436
	0.071429	0.317568	0.27957	0.188235	0.432225	0.010278	0.053649	0.821429	0	1	0.542986
	0	0.27027	0.376344	0.129412	0.20844	0.057923	0.089822	0.035714	0	0	0.03942
	0	0.837838	0.408602	0.576471	0.02046	0.094125	0.049588	0.035714	0	0	0.094209
	0.071429	0.189189	0.268817	0.411765	0.538363	0.091315	0.32868	0	0	0	0.193651
	0.571429	0.283784	0.290323	0.023529	0.028133	0.581772	0.07302	0.25	0	0	0.433454
	0.285714	0.135135	0.784946	0.411765	0.017903	0.995353	0.049325	0.017857	1	0	0.347773
	0	0.398649	0.387097	0.447059	0.012788	0.697836	0.01329	0.857143	0	1	0.660992
	0.142857	0.5	0.655914	0.423529	0.066496	0.065789	0.308087	0.446429	1	0	0.416886
	0.071429	0.337838	0.268817	0.364706	0.176471	0.23474	0.352027	0.357143	0	0	0.309801
	0	0.304054	0.731183	0.317647	0.034527	0.032387	0.028805	0.089286	0	0	0.046019
	0.642857	0.351351	0.301075	0.305882	0.028133	0.08929	0.291944	0.089286	0	0	0.38558
	0.071429	0.47973	0.430108	0.435294	0.053708	0.960777	0.056149	0.392857	1	1	0.548734
	0	0.324324	0.483871	0.317647	0.167519	0.683348	0.004659	0	0	0	0.117651
	0.071429	0.702703	0.419355	0.270588	0.031969	0.861309	0.080267	0.232143	0	0	0.365581
	0.5	0.743243	0.419355	0.035294	0.314578	0.470024	0.012532	0	0	0	0.457425
	0.571429	0.006757	0.688172	0.282353	0.002558	0.09215	0.09721	0.071429	0	0	0.187647
	0.642857	0.013514	0.55914	0.494118	0.095908	0.02167	0.187138	0	1	0	0.302171
	0	0.675676	0.333333	0.235294	0.225064	0.618622	0.030802	0.035714	0	0	0.163509
	0.071429	0.047297	0.655914	0.211765	0.170077	0.041787	0.017506	0.017857	0	0	0.037293
	0	0.709459	0.548387	0.129412	0.012788	0.544595	0.003142	0.089286	0	0	0.104943
	0	0.52027	0.666667	0	0.189258	0.588056	0.046381	0.089286	0	0	0.141839
	0	0.304054	0.698925	0.552941	0.251918	0.02784	0.0408	0	0	0	0.0804
	0	0.418919	0.365591	0.282353	0.016624	0.049923	0.342144	0.053571	0	0	0.059147
	0.285714	0.364865	0.27957	0.329412	0.085678	0.569583	0.073975	0.25	0	0	0.354043
	0	0.060811	0.688172	0.447059	0.023018	0.050742	0.067163	0.160714	0	0	0.054536

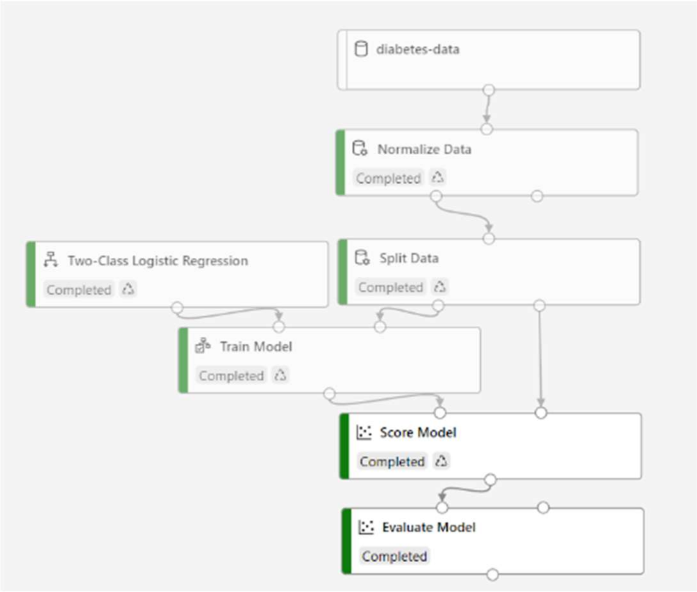
Scored Probabilities

Statistics

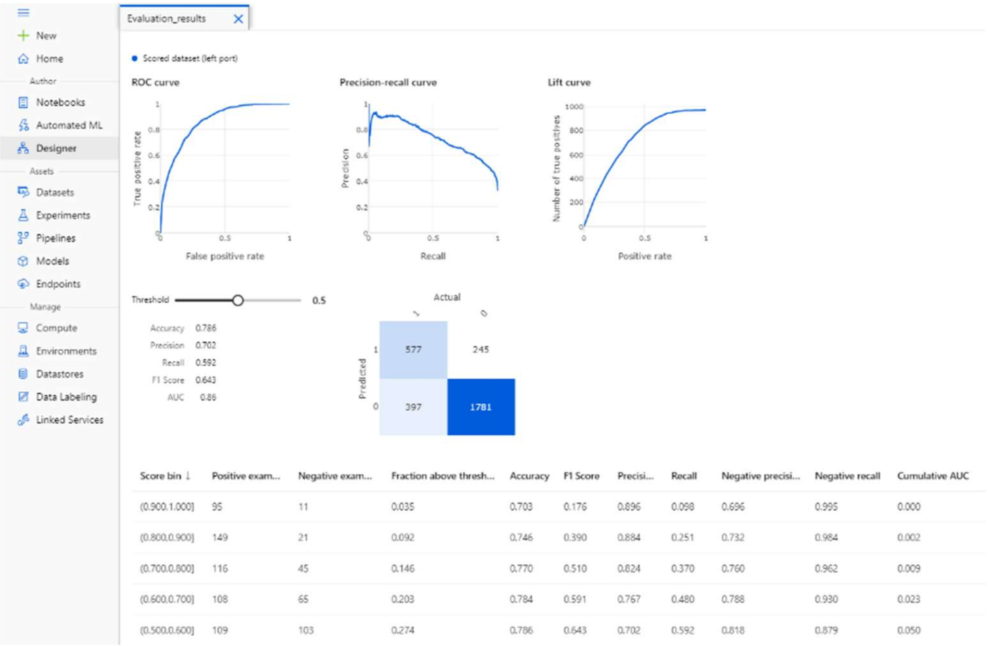
Mean	0.3308
Median	0.2341
Min	0.0121
Max	0.9988
Standard deviation	0.2753
Unique values	3000
Missing values	0
Feature type	Numeric Score

Visualizations

6. Add an evaluate model



7. View Result

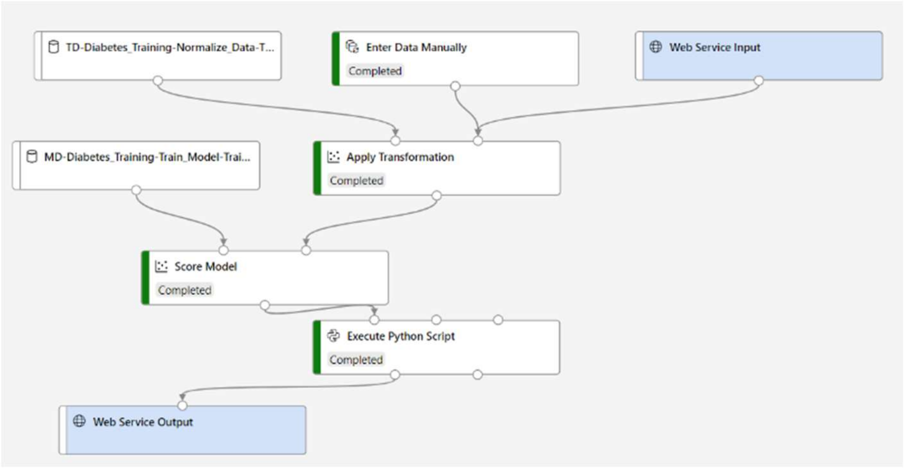


Accuracy	0.786
Precision	0.702
Recall	0.592
F1 Score	0.643
AUC	0.86

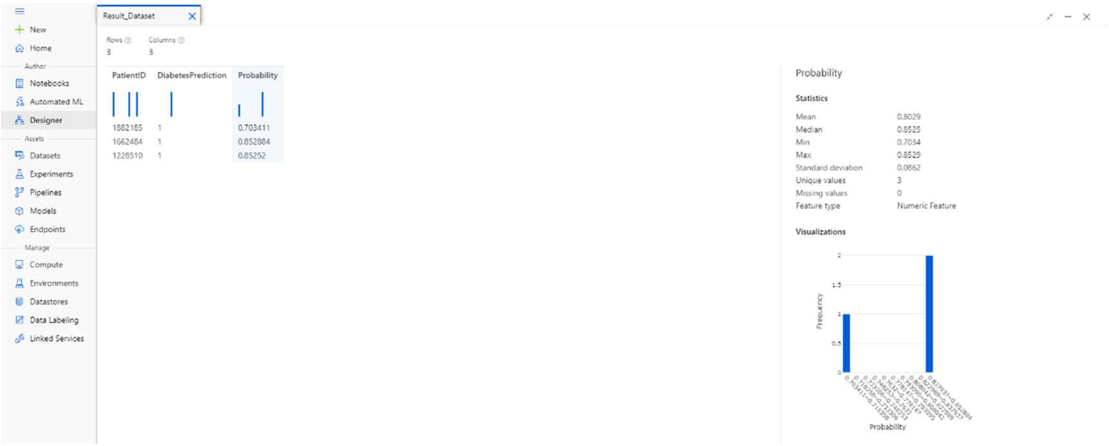
Predicted \ Actual	1	0
1	577	245
0	397	1781

Score bin ↓	Positive exam...	Negative exam...	Fraction above thresh...	Accuracy	F1 Score	Precisi...	Recall	Negative precisi...	Negative recall	Cumulative AUC
(0.900,1.000]	95	11	0.035	0.703	0.176	0.896	0.098	0.696	0.995	0.000
(0.800,0.900]	149	21	0.092	0.746	0.390	0.884	0.251	0.732	0.984	0.002
(0.700,0.800]	116	45	0.146	0.770	0.510	0.824	0.370	0.760	0.962	0.009
(0.600,0.700]	108	65	0.203	0.784	0.591	0.767	0.480	0.788	0.930	0.023
(0.500,0.600]	109	103	0.274	0.786	0.643	0.702	0.592	0.818	0.879	0.050

8. Create an inference pipeline



9. Run the pipeline & View result



Rows 3 Columns 3

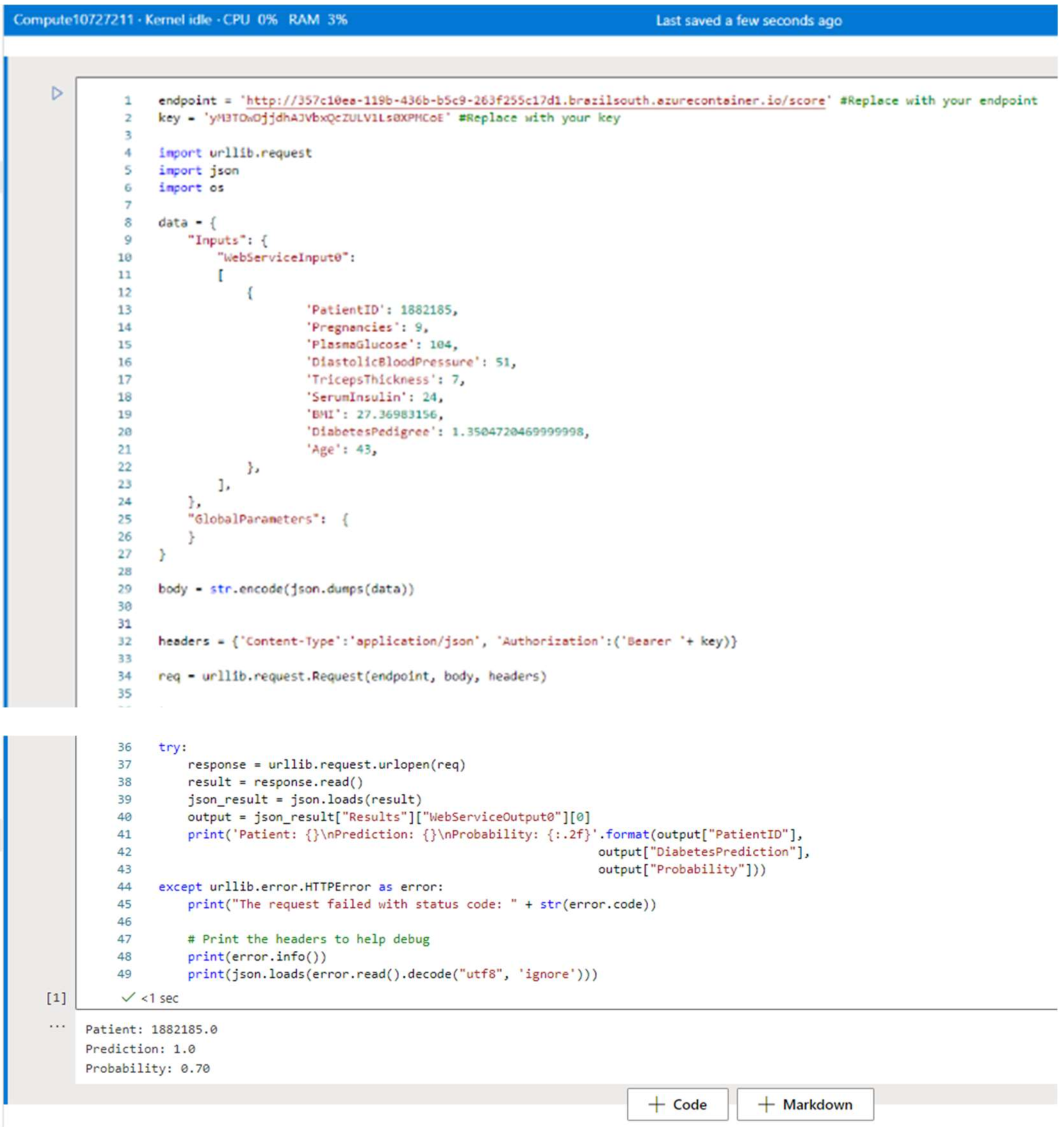
PatientID	DiabetesPrediction	Probability
1882185	1	0.703411
1662484	1	0.852884
1228510	1	0.85252

Probability

Statistics

Mean	0.8029
Median	0.8525
Min	0.7034
Max	0.8529
Standard deviation	0.0862
Unique values	3
Missing values	0
Feature type	Numeric Feature

10. Deploy a service & Test the service



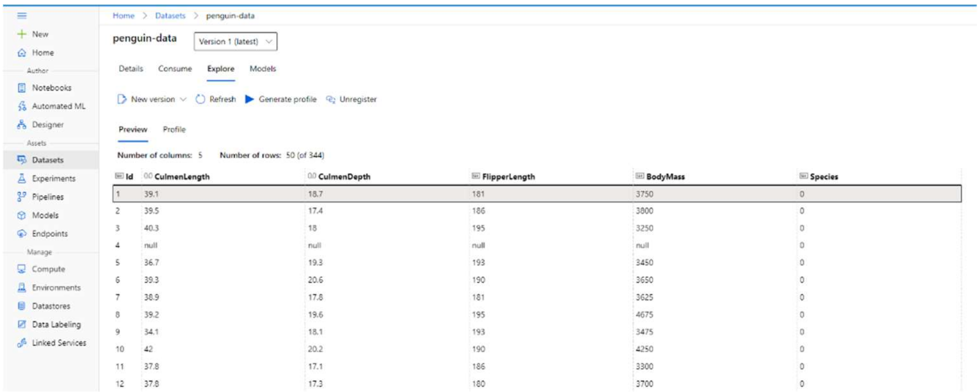
Compute10727211 · Kernel idle · CPU 0% RAM 3%
Last saved a few seconds ago

```python
endpoint = 'http://357c10ea-119b-436b-b5c9-263f255c17d1.brazilsouth.azurecontainer.io/score' #Replace with your endpoint
key = 'yM3TOwOjjdhAJVbxQcZULV1Ls0XPMCoE' #Replace with your key

import urllib.request
import json
import os

data = {
    "Inputs": {
        "WebServiceInput0":
        [
            {
                    'PatientID': 1882185,
                    'Pregnancies': 9,
                    'PlasmaGlucose': 104,
                    'DiastolicBloodPressure': 51,
                    'TricepsThickness': 7,
                    'SerumInsulin': 24,
                    'BMI': 27.36983156,
                    'DiabetesPedigree': 1.3504720469999998,
                    'Age': 43,
            },
        ],
    },
    "GlobalParameters":  {
    }
}

body = str.encode(json.dumps(data))


headers = {'Content-Type':'application/json', 'Authorization':('Bearer '+ key)}

req = urllib.request.Request(endpoint, body, headers)

try:
    response = urllib.request.urlopen(req)
    result = response.read()
    json_result = json.loads(result)
    output = json_result["Results"]["WebServiceOutput0"][0]
    print('Patient: {}\nPrediction: {}\nProbability: {:.2f}'.format(output["PatientID"],
                                                        output["DiabetesPrediction"],
                                                        output["Probability"]))
except urllib.error.HTTPError as error:
    print("The request failed with status code: " + str(error.code))

    # Print the headers to help debug
    print(error.info())
    print(json.loads(error.read().decode("utf8", 'ignore')))
```

```
Patient: 1882185.0
Prediction: 1.0
Probability: 0.70
```

Model 4:Create a Clustering model with Azure Machine Learning designer

1. Create a dataset

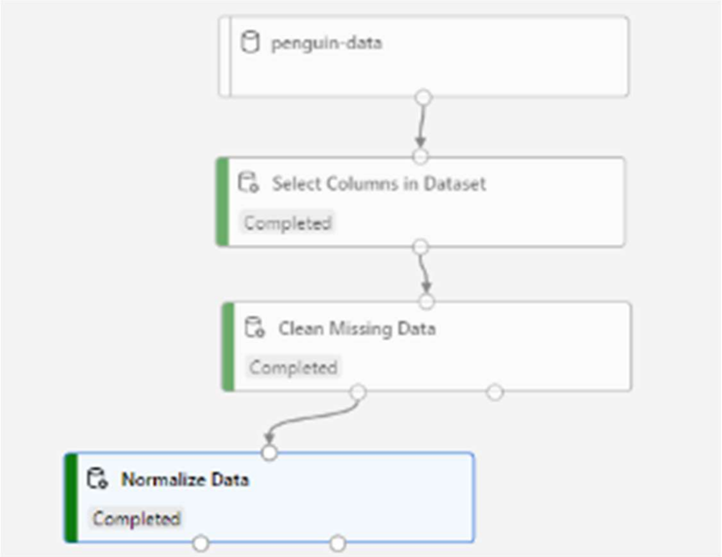


penguin-data

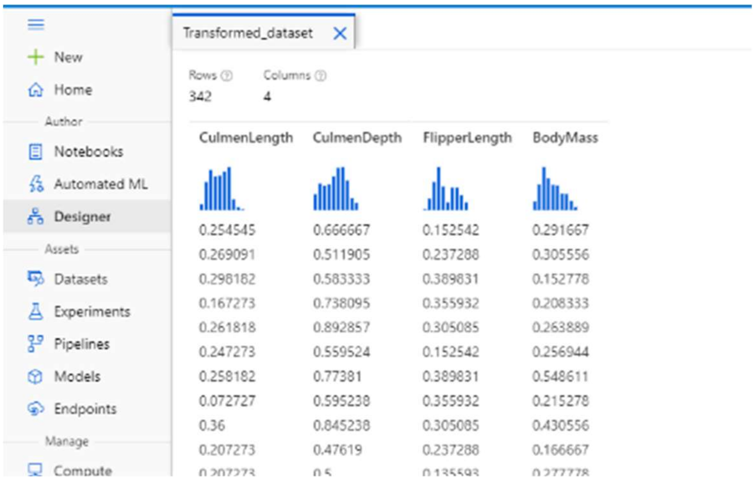
Number of columns: 5 Number of rows: 50 (of 344)

Id	CulmenLength	CulmenDepth	FlipperLength	BodyMass	Species
1	39.1	18.7	181	3750	0
2	39.5	17.4	186	3800	0
3	40.3	18	195	3250	0
4	null	null	null	null	0
5	36.7	19.3	193	3450	0
6	39.3	20.6	190	3650	0
7	38.9	17.8	181	3625	0
8	39.2	19.6	195	4675	0
9	34.1	18.1	193	3475	0
10	42	20.2	190	4250	0
11	37.8	17.1	186	3300	0
12	37.8	17.3	180	3700	0

2. Create a pipeline & Apply Transformations



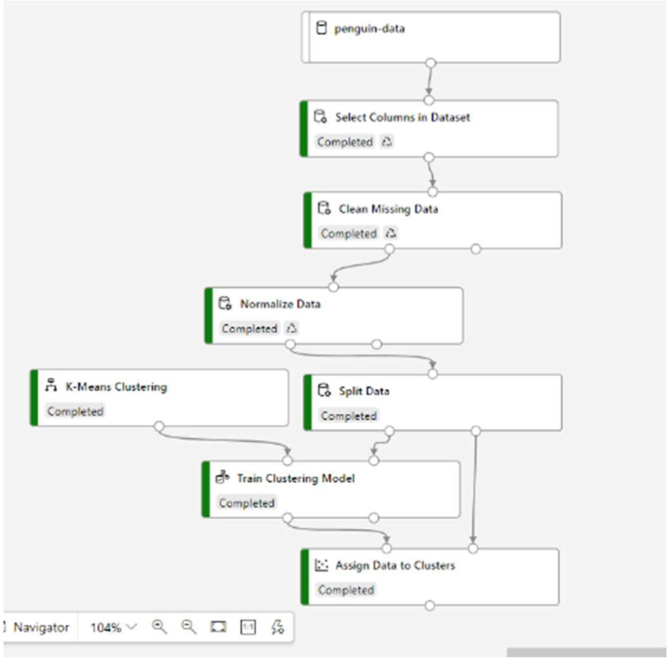
3. View the transformed data



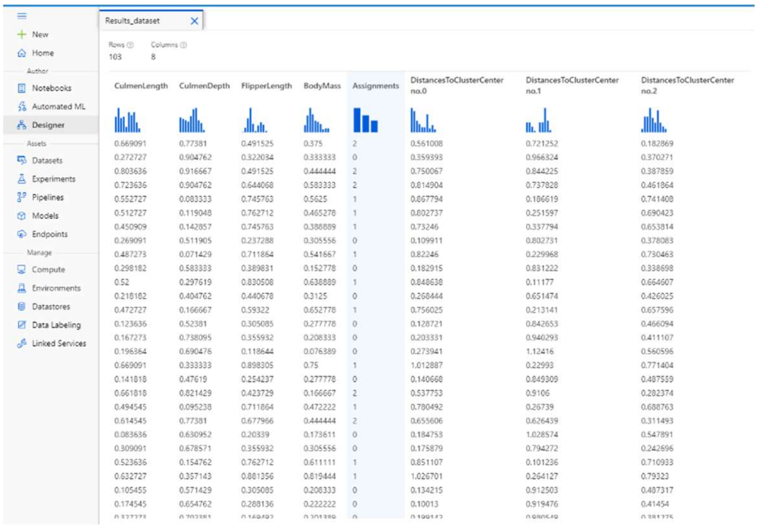
Rows 342 Columns 4

CulmenLength	CulmenDepth	FlipperLength	BodyMass
0.254545	0.666667	0.152542	0.291667
0.269091	0.511905	0.237288	0.305556
0.298182	0.583333	0.389831	0.152778
0.167273	0.738095	0.355932	0.208333
0.261818	0.892857	0.305085	0.263889
0.247273	0.559524	0.152542	0.256944
0.258182	0.77381	0.389831	0.548611
0.072727	0.595238	0.355932	0.215278
0.36	0.845238	0.305085	0.430556
0.207273	0.47619	0.237288	0.166667
0.207273	0.5	0.135593	0.277778

4. Add training models



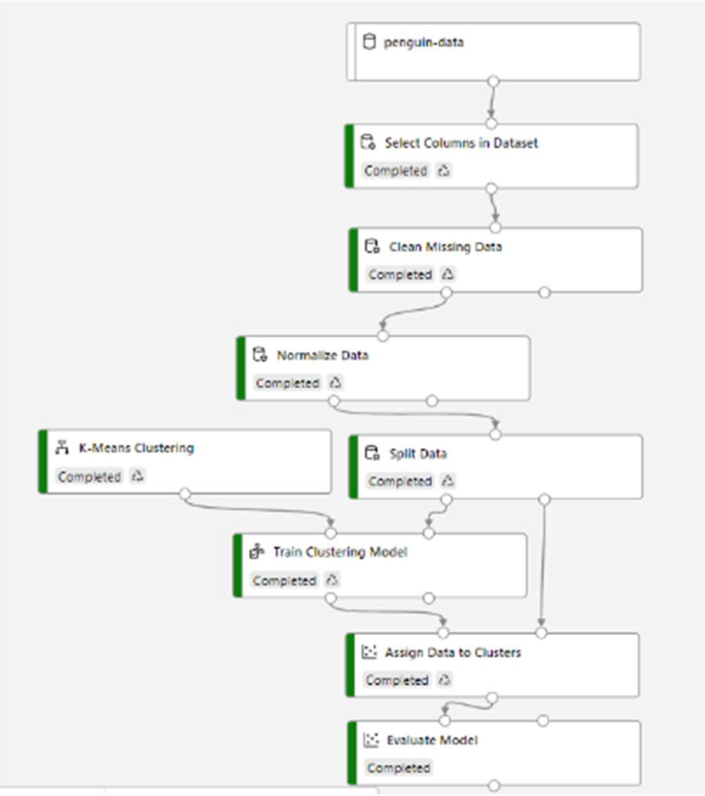
5. Run the training pipeline & View results



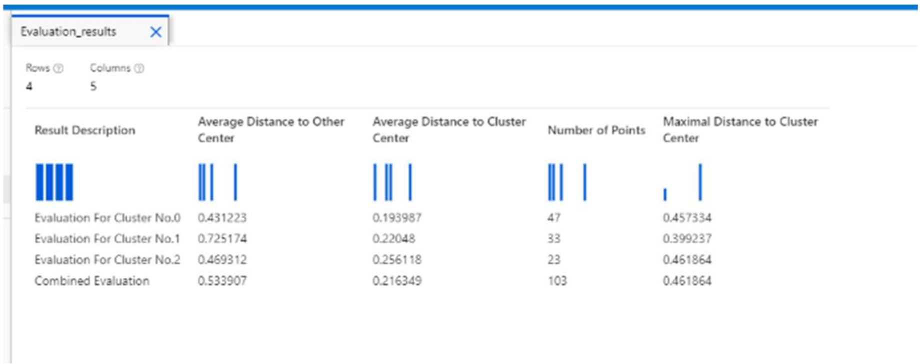
Rows 103 Columns 8

CulmenLength	CulmenDepth	FlipperLength	BodyMass	Assignments	DistancesToClusterCenter no.0	DistancesToClusterCenter no.1	DistancesToClusterCenter no.2
0.669091	0.77381	0.491525	0.375	2	0.561008	0.721252	0.182869
0.272727	0.904762	0.322034	0.333333	0	0.359393	0.966324	0.370271
0.803636	0.916667	0.491525	0.444444	2	0.750067	0.844225	0.387859
0.723636	0.904762	0.644068	0.583333	2	0.814904	0.737828	0.461864
0.552727	0.083333	0.745763	0.5625	1	0.867794	0.186619	0.741408
0.512727	0.119048	0.762712	0.465278	1	0.802737	0.251597	0.690423
0.450909	0.142857	0.745763	0.388889	1	0.73246	0.337794	0.653814
0.269091	0.511905	0.237288	0.305556	0	0.109911	0.802731	0.378083
0.487273	0.071429	0.711864	0.541667	1	0.82246	0.229968	0.730463
0.298182	0.583333	0.389831	0.152778	0	0.182915	0.831222	0.338698
0.52	0.297619	0.830508	0.638889	1	0.848638	0.11177	0.664607
0.218182	0.404762	0.440678	0.3125	0	0.268444	0.651474	0.426025
0.472727	0.166667	0.59322	0.652778	1	0.756025	0.213141	0.657596
0.123636	0.52381	0.305085	0.277778	0	0.128721	0.842653	0.466094
0.167273	0.738095	0.355932	0.208333	0	0.203331	0.940293	0.411107
0.196364	0.690476	0.118644	0.076389	0	0.273941	1.12416	0.560596
0.669091	0.333333	0.898305	0.75	1	1.012887	0.22993	0.771404
0.141818	0.47619	0.254237	0.277778	0	0.140668	0.849309	0.487559
0.661818	0.821429	0.423729	0.166667	2	0.537753	0.9106	0.282374
0.494545	0.095238	0.711864	0.472222	1	0.780492	0.26739	0.688763
0.614545	0.77381	0.677966	0.444444	2	0.655606	0.626439	0.311493
0.083636	0.630952	0.20339	0.173611	0	0.184753	1.028574	0.547891
0.309091	0.678571	0.355932	0.305556	0	0.175879	0.794272	0.242696
0.523636	0.154762	0.762712	0.611111	1	0.851107	0.101236	0.710933
0.632727	0.357143	0.881356	0.819444	1	1.026701	0.264127	0.79323
0.105455	0.571429	0.305085	0.208333	0	0.134215	0.912503	0.487317
0.174545	0.654762	0.288136	0.222222	0	0.10013	0.919476	0.41454
0.327273	0.702381	0.169492	0.201389	0	0.199142	0.980549	0.381275

6. Add and Evaluate Model



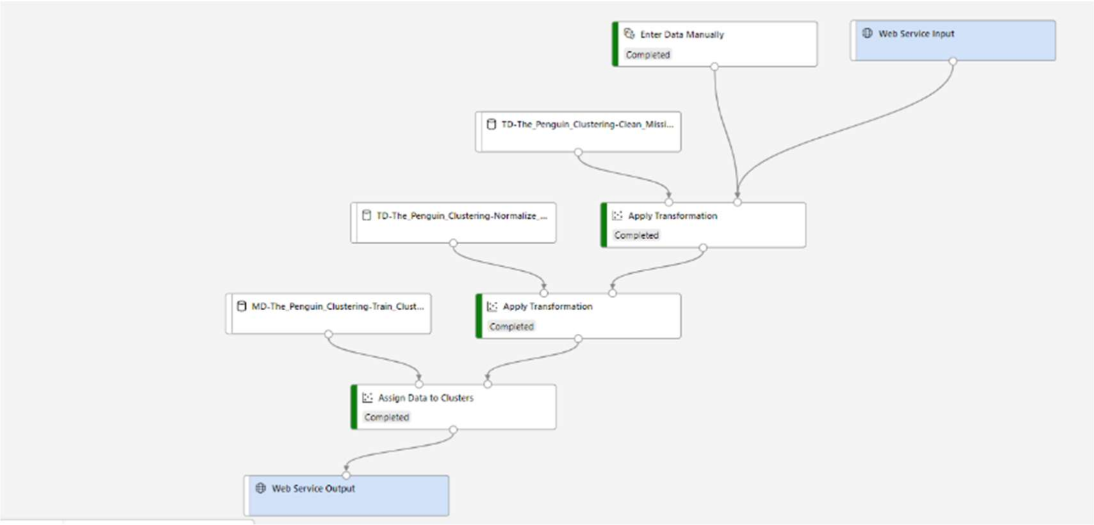
7. View Result



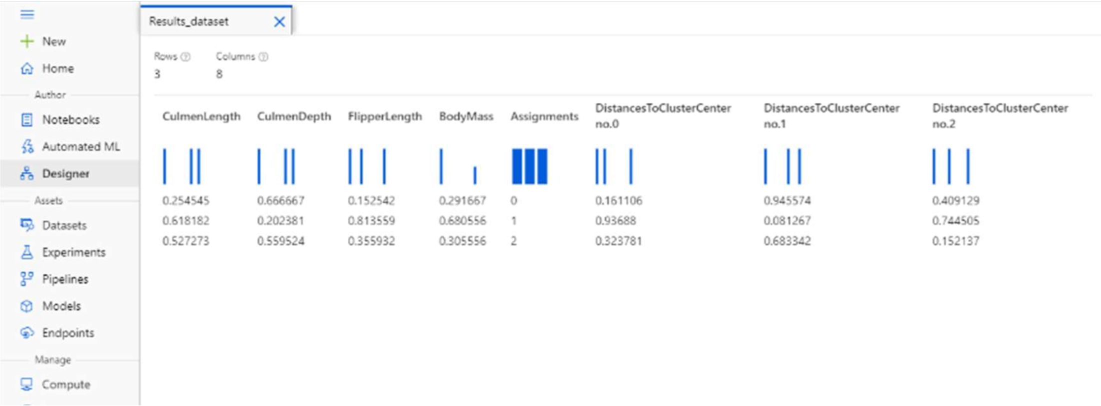
Rows 4 Columns 5

Result Description	Average Distance to Other Center	Average Distance to Cluster Center	Number of Points	Maximal Distance to Cluster Center
Evaluation For Cluster No.0	0.431223	0.193987	47	0.457334
Evaluation For Cluster No.1	0.725174	0.22048	33	0.399237
Evaluation For Cluster No.2	0.469312	0.256118	23	0.461864
Combined Evaluation	0.533907	0.216349	103	0.461864

8. Create an inference pipeline



9. View Result

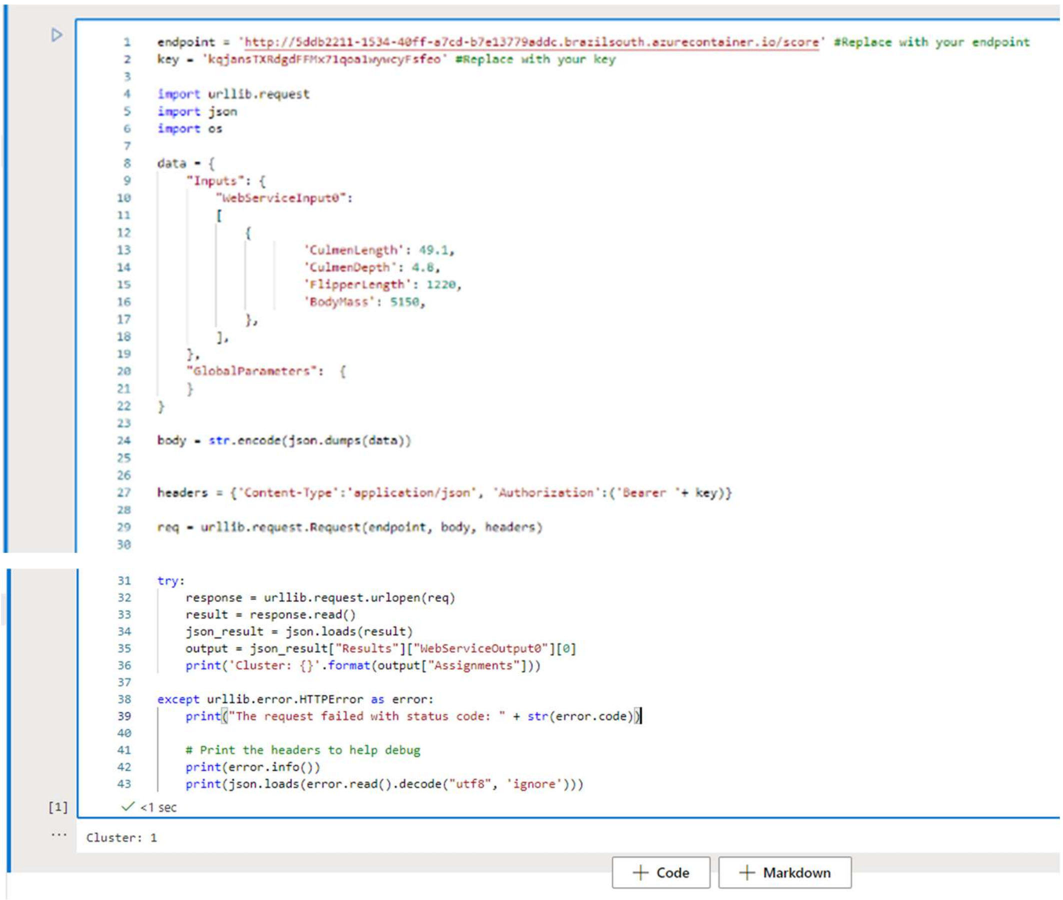


Results_dataset

Rows 3 Columns 8

CulmenLength	CulmenDepth	FlipperLength	BodyMass	Assignments	DistancesToClusterCenter no.0	DistancesToClusterCenter no.1	DistancesToClusterCenter no.2
0.254545	0.666667	0.152542	0.291667	0	0.161106	0.945574	0.409129
0.618182	0.202381	0.813559	0.680556	1	0.93688	0.081267	0.744505
0.527273	0.559524	0.355932	0.305556	2	0.323781	0.683342	0.152137

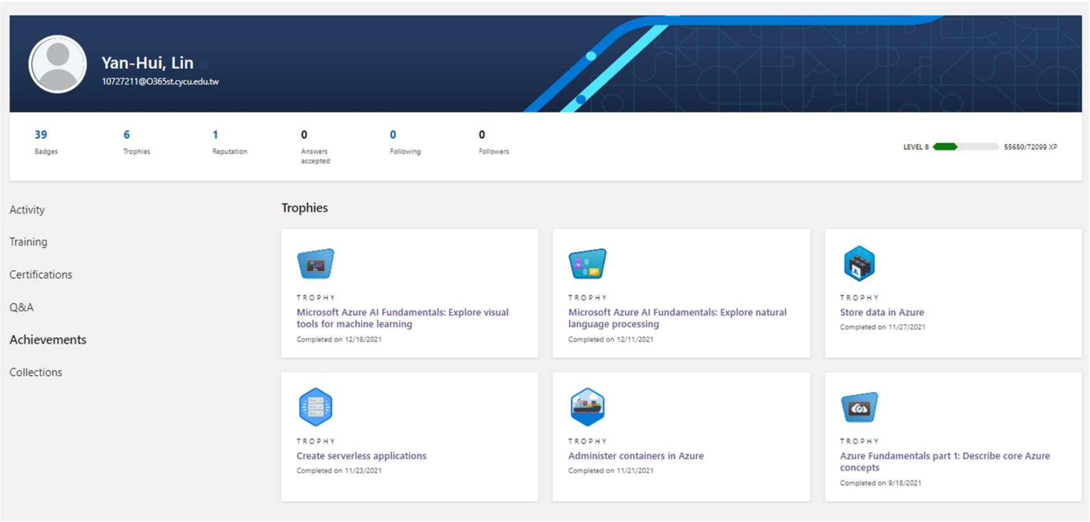
10. Deploy a service and test the service



```
endpoint = 'http://5ddb2211-1534-40ff-a7cd-b7e13779addc.brazilsouth.azurecontainer.io/score' #Replace with your endpoint
key = 'kqjansTXRdgdFFMx71qoalwywcyFsfeo' #Replace with your key

import urllib.request
import json
import os

data = {
    "Inputs": {
        "WebServiceInput0":
        [
            {
                    'CulmenLength': 49.1,
                    'CulmenDepth': 4.8,
                    'FlipperLength': 1220,
                    'BodyMass': 5150,
            },
        ],
    },
    "GlobalParameters":  {
    }
}

body = str.encode(json.dumps(data))


headers = {'Content-Type':'application/json', 'Authorization':('Bearer '+ key)}

req = urllib.request.Request(endpoint, body, headers)


try:
    response = urllib.request.urlopen(req)
    result = response.read()
    json_result = json.loads(result)
    output = json_result["Results"]["WebServiceOutput0"][0]
    print('Cluster: {}'.format(output["Assignments"]))

except urllib.error.HTTPError as error:
    print("The request failed with status code: " + str(error.code))

    # Print the headers to help debug
    print(error.info())
    print(json.loads(error.read().decode("utf8", 'ignore')))
```

```
Cluster: 1
```

Take screenshots of Badges and Trophies



Yan-Hui, Lin
10727211@O365st.cycu.edu.tw

39 Badges
6 Trophies
1 Reputation
0 Answers accepted
0 Following
0 Followers

LEVEL 8 55650/72099 XP

Trophies

TROPHY
Microsoft Azure AI Fundamentals: Explore visual tools for machine learning
Completed on 12/18/2021

TROPHY
Microsoft Azure AI Fundamentals: Explore natural language processing
Completed on 12/11/2021

TROPHY
Store data in Azure
Completed on 11/27/2021

TROPHY
Create serverless applications
Completed on 11/23/2021

TROPHY
Administer containers in Azure
Completed on 11/21/2021

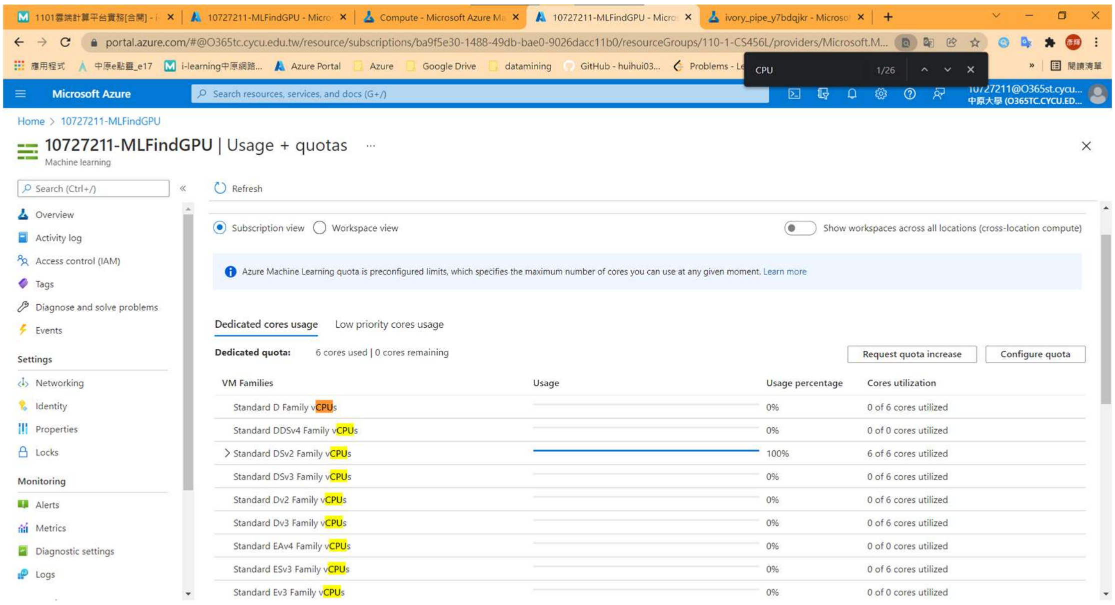
TROPHY
Azure Fundamentals part 1: Describe core Azure concepts
Completed on 9/18/2021

Learned from the Learning Path

Azure Machine Learning 在機器學習中提供相當多的參數，從計算資源、演算法、GUI 介面、部署方式、計算模型、驗證模型到分析模型，Azure 的 ML 提供的計算相當多元，不管是資料前處理、模型訓練、驗證、甚至圖表分析都有 Model 支援，且分析的結果數據相當多元。Pipeline 的 GUI 概念讓開發人員可以更專注於他們的開發工作，Learning Path 過程中也從中習得不少 ML 的演算法知識與驗證觀念。

Problems

1. 部分的 Region 可能資源不夠，Run Model 時可能因 Training 時間過長而導致系統主動判斷 Timeout Error。
2. 可以考慮開啟 GPU 模式（所有的地域都無法看見可用 GPU 之資源）。


1101雲端計算平台實務[合開] - i
10727211-MLFindGPU - Micro
Compute - Microsoft Azure Ma
10727211-MLFindGPU - Micro
ivory_pipe_y7bdqjkr - Microsof
portal.azure.com/#@O365tc.cycu.edu.tw/resource/subscriptions/ba9f5e30-1488-49db-bae0-9026dacc11b0/resourceGroups/110-1-CS456L/providers/Microsoft.M...
應用程式
中原e點靈_e17
i-learning中原網路...
Azure Portal
Azure
Google Drive
datamining
GitHub - huihui03...
Problems - Le
CPU
1/26
閱讀清單
Microsoft Azure
Search resources, services, and docs (G+/)
10727211@O365st.cycu...
中原大學 (O365TC.CYCU.ED...
Home > 10727211-MLFindGPU
Search (Ctrl+/)
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Events
Settings
Networking
Identity
Properties
Locks
Monitoring
Alerts
Metrics
Diagnostic settings
Logs
Refresh
Request quota increase
Configure quota

10727211-MLFindGPU | Usage + quotas

Machine learning

● Subscription view ○ Workspace view

Show workspaces across all locations (cross-location compute)

Azure Machine Learning quota is preconfigured limits, which specifies the maximum number of cores you can use at any given moment. Learn more

Dedicated cores usage   Low priority cores usage

Dedicated quota: 6 cores used | 0 cores remaining

VM Families	Usage	Usage percentage	Cores utilization
Standard D Family vCPUs		0%	0 of 6 cores utilized
Standard DDSv4 Family vCPUs		0%	0 of 0 cores utilized
> Standard DSv2 Family vCPUs		100%	6 of 6 cores utilized
Standard DSv3 Family vCPUs		0%	0 of 6 cores utilized
Standard Dv2 Family vCPUs		0%	0 of 6 cores utilized
Standard Dv3 Family vCPUs		0%	0 of 6 cores utilized
Standard EAv4 Family vCPUs		0%	0 of 0 cores utilized
Standard ESv3 Family vCPUs		0%	0 of 6 cores utilized
Standard Ev3 Family vCPUs		0%	0 of 0 cores utilized

FeedBack

透過 Azure Machine Learning 的 Pipeline，從中體認 Azure 對於「No Code」支援的強大，使用者可以將實作需要的理論透過 GUI Block 的方式實作想要的應用，且這些 GUI 也支援防呆機制，在 Pipeline IO 加入審查機制，讓使用者可以減少 Debug 的過程，Azure 提供的演算法也相當多元，最後也可以透過 GUI 將 Model 做部署至 Container Instance 供及時 Inference、Predict，供對於機器學習的初心者而言，Azure Machine Learn 是一個相當適合的環境。