AI on Cloud

depa Microsoft AIT Asian Institute of Technology SCB ACADEMY

AI on Cloud

Chapter 2

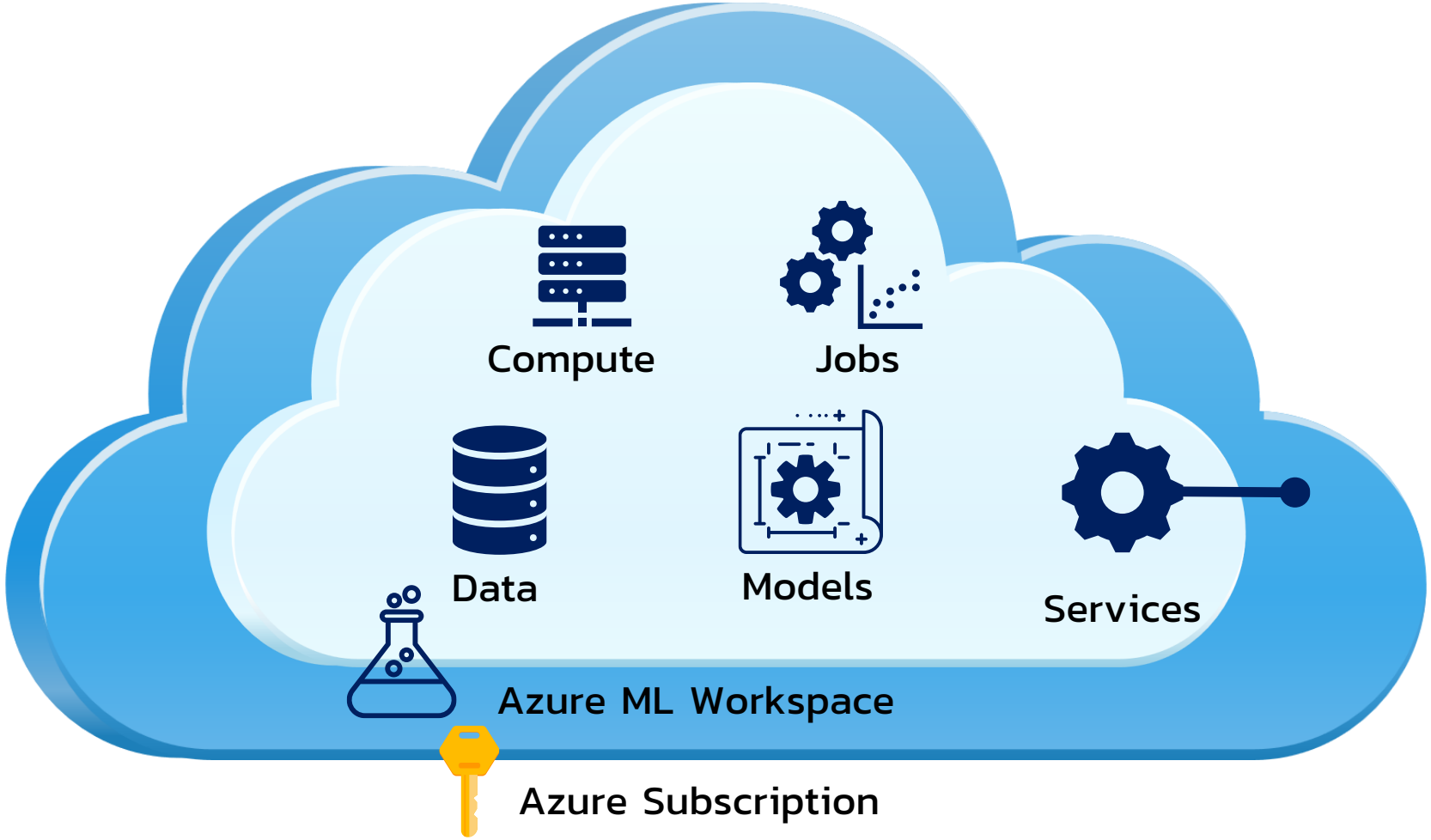
การเรียนรู้ของเครื่องจักร Machine Learning

depa Microsoft AIT Asian Institute of Technology SCB ACADEMY

Lab 2 สร้าง Regression Model ด้วย Azure Machine Learning Designer

What is Azure Machine Learning?

A cloud-based platform for machine learning



Machine learning in Microsoft Azure

Azure Machine Learning provides the following features and capabilities:

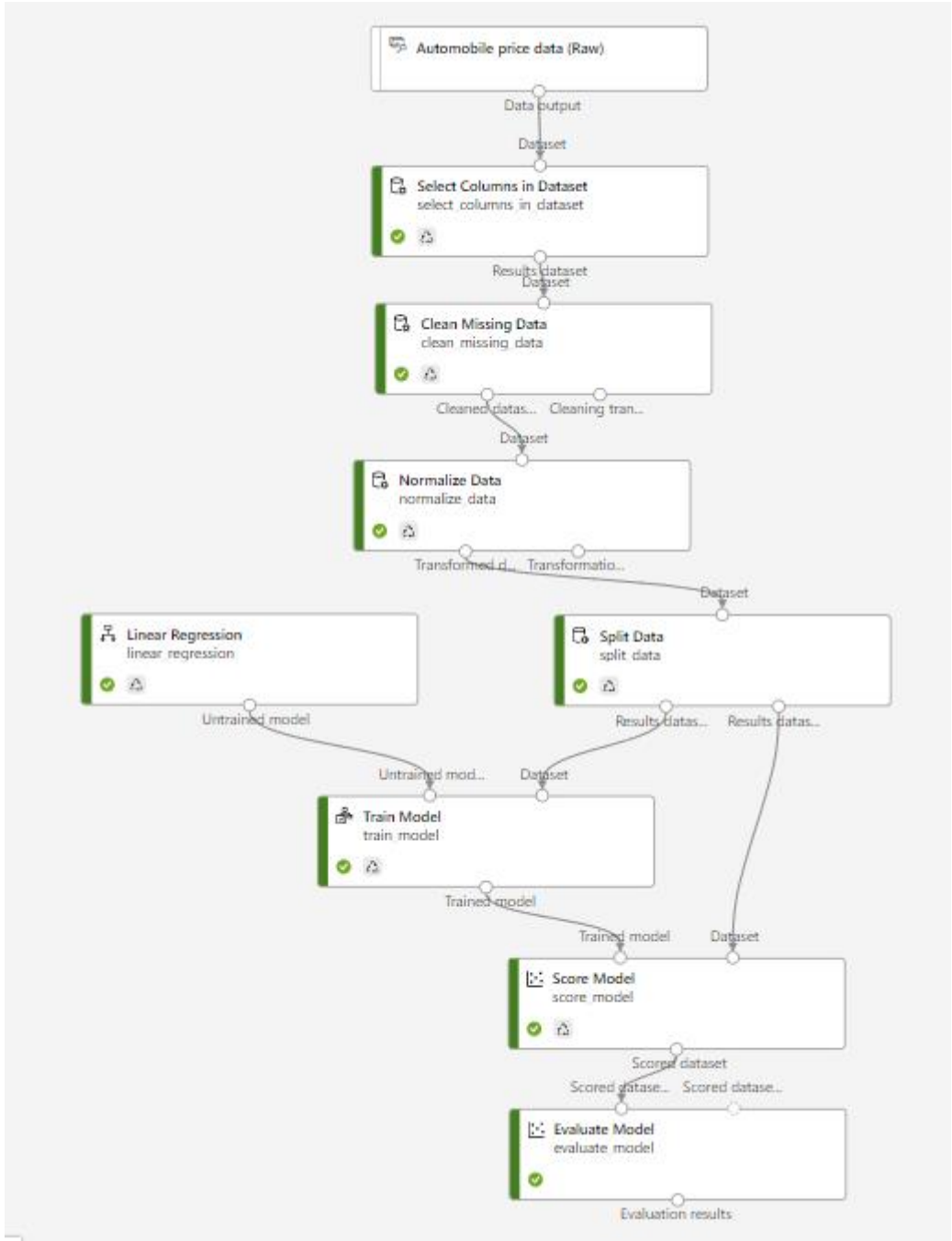
Feature	Capability
Automated machine learning	This feature enables non-experts to quickly create an effective machine learning model from data.
Azure Machine Learning designer	A graphical interface enabling no-code development of machine learning solutions.
Data and compute management	Cloud-based data storage and compute resources that professional data scientists can use to run data experiment code at scale.
Pipelines	Data scientists, software engineers, and IT operations professionals can define pipelines to orchestrate model training, deployment, and management tasks.

https://docs.microsoft.com/en-us/learn/modules/get-started-ai-fundamentals/2-understand-machine-learn

Azure Machine Learning Designer

Visual tool for creating a machine learning *pipeline*

- **Part 1:** Use a *training pipeline* to train and evaluate a model
- **Part 2:** Create an *inference pipeline* to predict labels from new data
- **Part 3**: Deploy the inference pipeline as a *service* for apps to use



Lab2: สร้าง Regression Model ด้วย Azure Machine Learning Designer

เป้าหมาย: เรียนรู้การใช้ Azure Machine Learning Designer เพื่อ

- สร้างและ Run Training Pipeline
- สร้าง Inference Pipeline
- Deploy Predictive Model

- Tool: Azure ML Designer
- Model: Regression Model
- Metric: MAE, RMSE, RSE, RAE, R2

Regression

เป็น Supervised machine learning technique ใช้สำหรับ Predict numeric values

Dataset ควรประกอบด้วย:

- **Features** ➔ Characteristics of the entity for which you want to make a prediction.
- **Label** ➔ The numeric value that you want to train a model to predict

Case Study และ ข้อมูลที่ใช้

- **Case Study**: บริษัทขายรถยนต์ต้องการพยากรณ์ราคาขายรถยนต์โดยใช้คุณลักษณะของรถยนต์
  - ***Features*** → Characteristics (ได้แก่ make, engine size, number of seats, etc)
  - ***Label*** → ราคา (price)

ตัวอย่างข้อมูลและ features

make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location
audi	gas	std	two	sedan	fwd	front
audi	gas	std	four	sedan	fwd	front
audi	gas	std	four	wagon	fwd	front
audi	gas	turbo	four	sedan	fwd	front
audi	gas	turbo	two	hatchback	4wd	front
bmw	gas	std	two	sedan	rwd	front
bmw	gas	std	four	sedan	rwd	front
bmw	gas	std	two	sedan	rwd	front

Part 1

Use a Training Pipeline to train and evaluate a model


Step 1: สร้าง Compute Instance

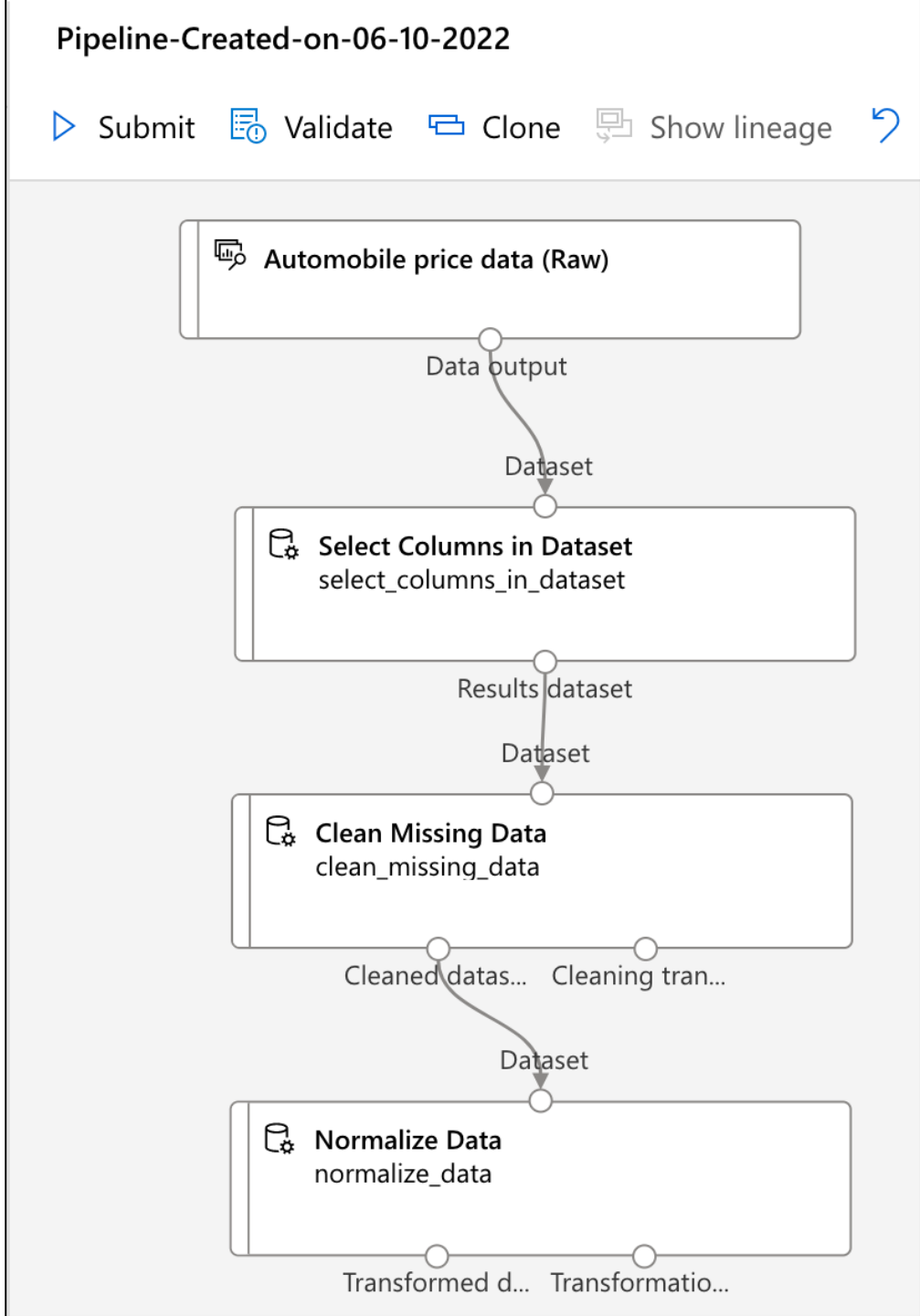

Step 2: สร้าง Compute Clusters


Step 3: สร้าง Pipeline สำหรับ
Manage Dataset

Step 4: ทำ Data Transformation เพื่อจัดการข้อมูลให้พร้อมสำหรับการสร้างโมเดล

- ทำความเข้าใจและสำรวจข้อมูล
- เลือก/Remove Features ที่ต้องการ/ไม่ต้องการใช้ในการสร้างโมเดล
- Clean Missing Values
- Normalize ข้อมูล ด้วยวิธี MinMax สำหรับ Features ต่อไปนี้

- Symbolling
- Wheel-base
- Length
- Width
- Height
- Curb-weight
- Engine-size
- Bore
- Stroke
- Compression-ratio
- Horsepower
- Peak-rpm
- City-mpg
- Highway-mpg



Step 5: Run the Pipeline และ: Explore ผลลัพธ์

- Dataset ได้ถูก Prepare และพร้อมสำหรับการสร้าง Model

Transformed_dataset

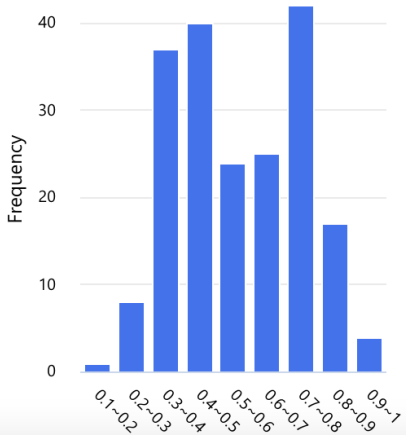
Rows 199 Columns 25

	height	curb-weight	engine-type	num-of-cylinders	engine-size	fuel-system	bore	stroke	compression-ratio	horsepowe
667	0.083333	0.411171	dohc	four	0.260377	mpfi	0.664286	0.290476	0.125	0.2625
667	0.083333	0.411171	dohc	four	0.260377	mpfi	0.664286	0.290476	0.125	0.2625
333	0.383333	0.517843	ohcv	six	0.343396	mpfi	0.1	0.666667	0.125	0.441667
667	0.541667	0.329325	ohc	four	0.181132	mpfi	0.464286	0.633333	0.1875	0.225
333	0.541667	0.518231	ohc	five	0.283019	mpfi	0.464286	0.633333	0.0625	0.279167
	0.441667	0.395268	ohc	five	0.283019	mpfi	0.464286	0.633333	0.09375	0.258333
	0.658333	0.525989	ohc	five	0.283019	mpfi	0.464286	0.633333	0.09375	0.258333
	0.658333	0.568658	ohc	five	0.283019	mpfi	0.464286	0.633333	0.09375	0.258333
	0.675	0.61986	ohc	five	0.264151	mpfi	0.421429	0.633333	0.08125	0.383333
333	0.35	0.60706	ohc	five	0.264151	mpfi	0.421429	0.633333	0	0.466667
	0.541667	0.351823	ohc	four	0.177358	mpfi	0.685714	0.347619	0.1125	0.220833
	0.541667	0.351823	ohc	four	0.177358	mpfi	0.685714	0.347619	0.1125	0.220833
	0.541667	0.474011	ohc	six	0.388679	mpfi	0.55	0.533333	0.125	0.304167
	0.541667	0.495345	ohc	six	0.388679	mpfi	0.55	0.533333	0.125	0.304167
	0.658333	0.607836	ohc	six	0.388679	mpfi	0.55	0.533333	0.125	0.304167

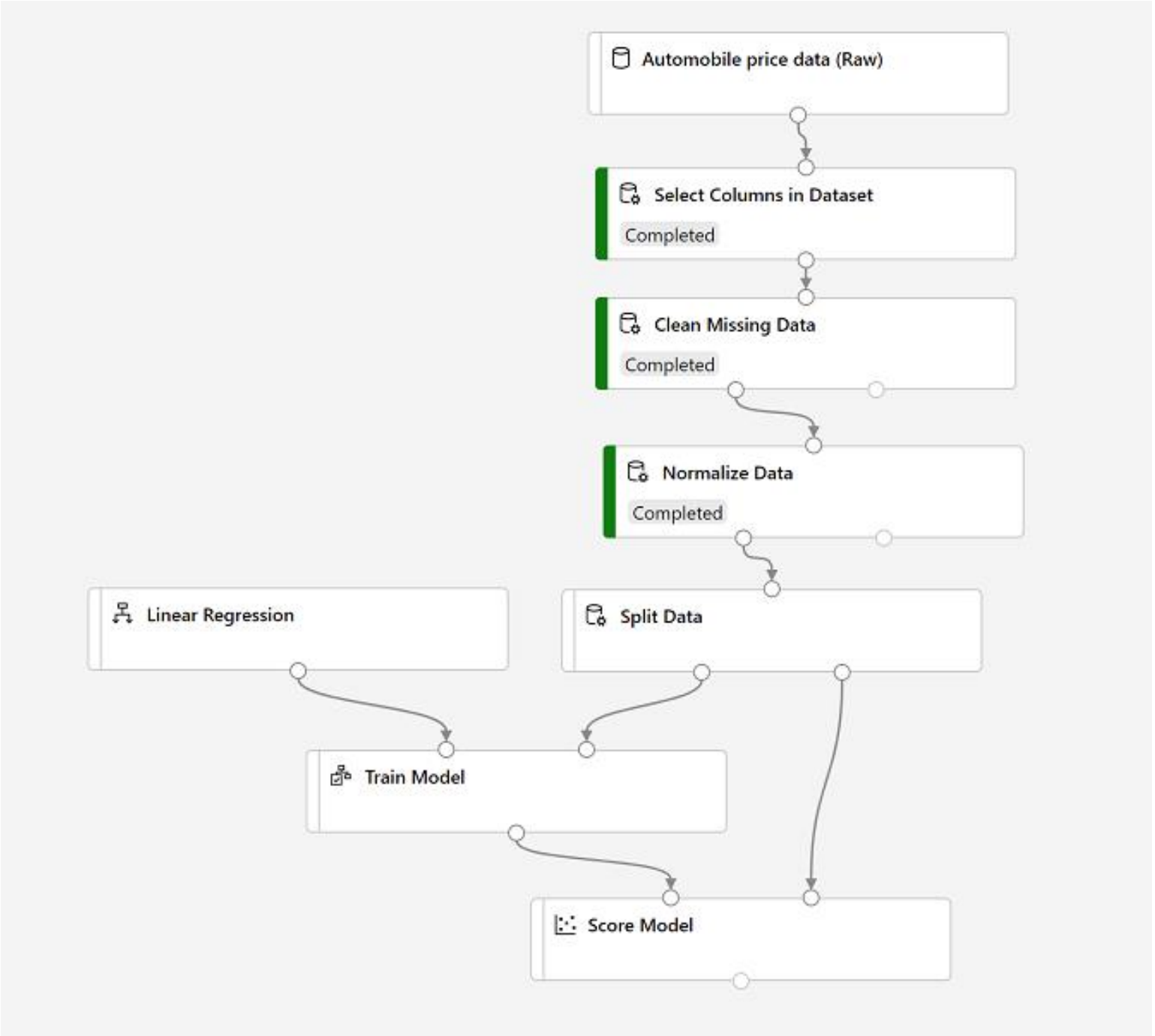
Statistics

Mean	0.5632
Median	0.55
Min	0
Max	1
Standard deviation	0.1961
Unique values	38
Missing values	0
Feature type	Numeric Feature

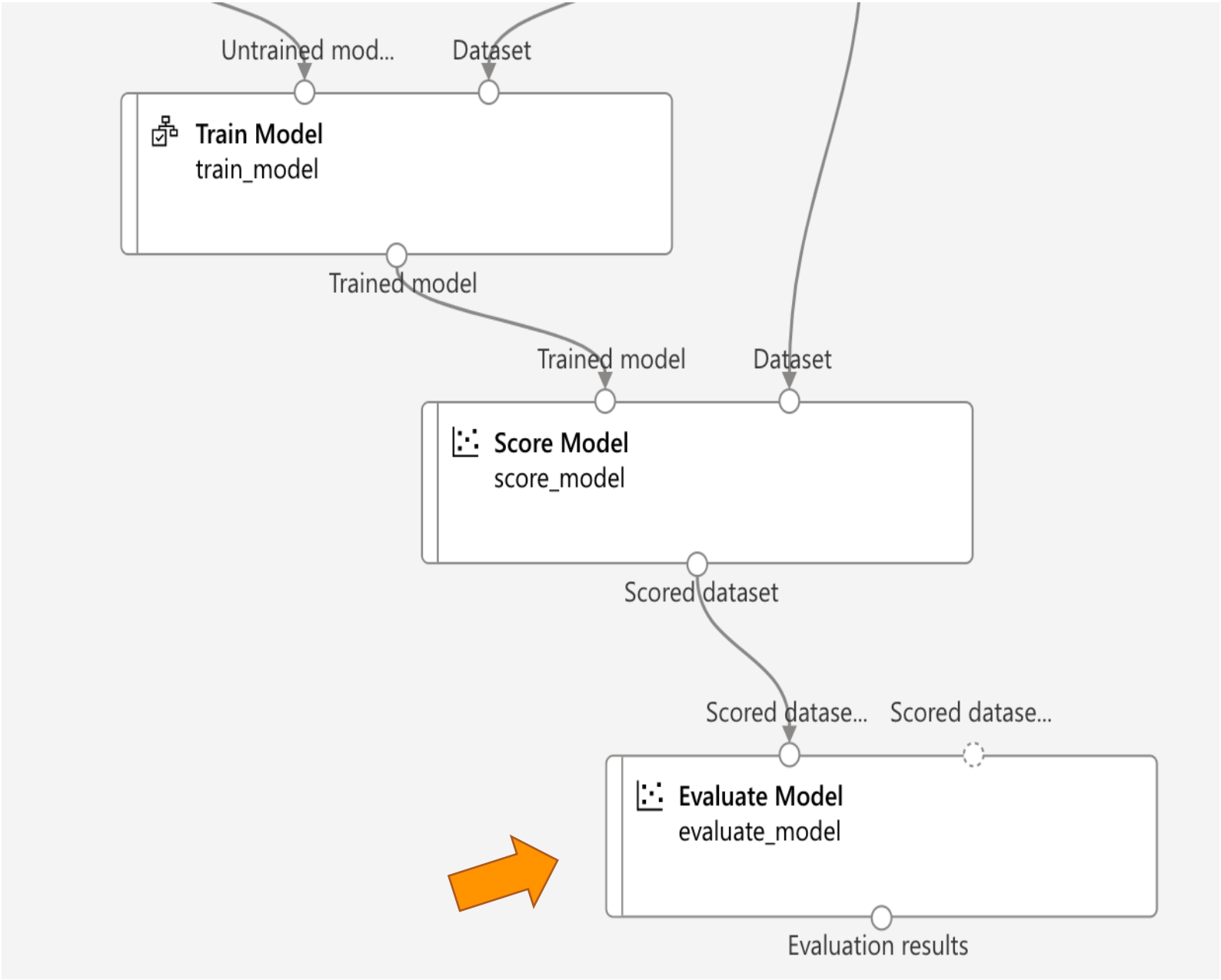
Visualizations



Step 6: Create Training Pipeline for Model Training



Step 7: Evaluate Model



Regression Performance Metrics

Metrics	Description
Mean Absolute Error (MAE)	The average difference between predicted values and true values. This value is based on the same units as the label, in this case dollars. The lower this value is, the better the model is predicting.
Root Mean Squared Error (RMSE)	The square root of the mean squared difference between predicted and true values. The result is a metric based on the same unit as the label (dollars). When compared to the MAE (above), a larger difference indicates greater variance in the individual errors (for example, with some errors being very small, while others are large).
Relative Squared Error (RSE)	A relative metric between 0 and 1 based on the square of the differences between predicted and true values. The closer to 0 this metric is, the better the model is performing. Because this metric is relative, it can be used to compare models where the labels are in different units.
Relative Absolute Error (RAE)	A relative metric between 0 and 1 based on the absolute differences between predicted and true values. The closer to 0 this metric is, the better the model is performing. Like RSE, this metric can be used to compare models where the labels are in different units.
Coefficient of Determination (R^2)	This metric is more commonly referred to as R-Squared, and summarizes how much of the variance between predicted and true values is explained by the model. The closer to 1 this value is, the better the model is performing.

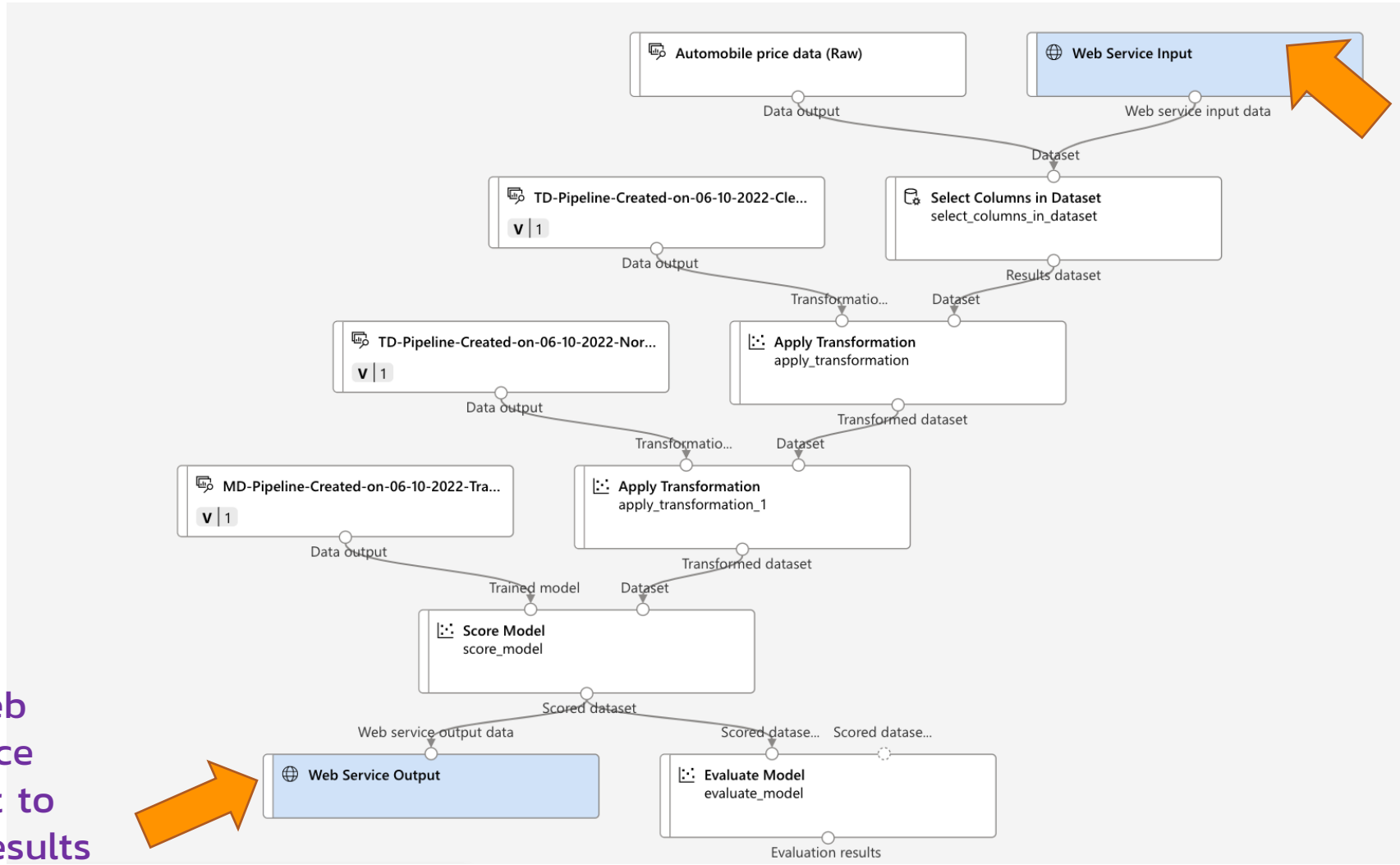
https://docs.microsoft.com/en-us/learn/modules/create-regression-model-azure-machine-learning-designer/6-evaluate-model

Part

2

Create an Inference Pipeline to predict labels from new data

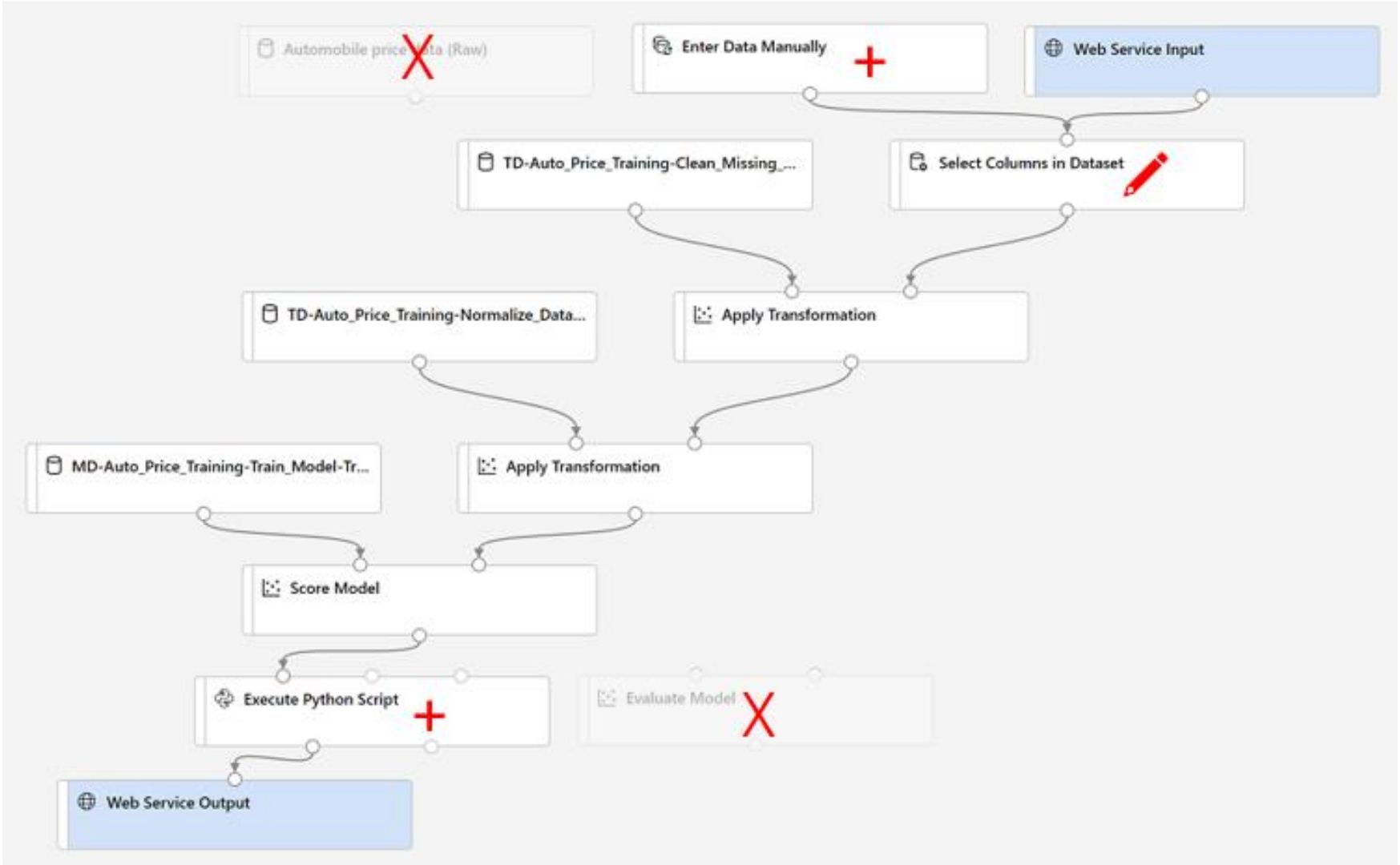
Step 1: สร้าง Inference Pipeline



A web service input for new data to be submitted

A web service output to return results

Step 2: แก้ไข The Inference Pipeline



Step 2.1 ลบ Automobile Price Data (Raw)

Step 2.2 เพิ่ม Enter Data Manually โดย Copy&Paste ข้อมูล CSV data ของ 3 Cars ดังนี้

```
symboling,normalized-losses,make,fuel-type,aspiration,num-of-doors,body-style,drive-wheels,engine-location,wheel-base,length,width,height,curb-weight,engine-type,num-of-cylinders,engine-size,fuel-system,bore,stroke,compression-ratio,horsepower,peak-rpm,city-mpg,highway-mpg
3,NaN,alfa-romero,gas,std,two,convertible,rwd,front,88.6,168.8,64.1,48.8,2548,dohc,four,130,mpfi,3.47,2.68,9,111,5000,21,27
3,NaN,alfa-romero,gas,std,two,convertible,rwd,front,88.6,168.8,64.1,48.8,2548,dohc,four,130,mpfi,3.47,2.68,9,111,5000,21,27
1,NaN,alfa-romero,gas,std,two,hatchback,rwd,front,94.5,171.2,65.5,52.4,2823,ohcv,six,152,mpfi,2.68,3.47,9,154,5000,19,26
```

Step 2.3 แก้ไข **Select Columns in Dataset** โดยลบ Column Price ออก

Step 2.4 ลบ **Evaluate Model** ออก

Step 2.5 เพิ่ม **Execute Python Script** เพื่อคืนค่าผลลัพธ์ Predicted Price เป็น Output ของ Web Service Output

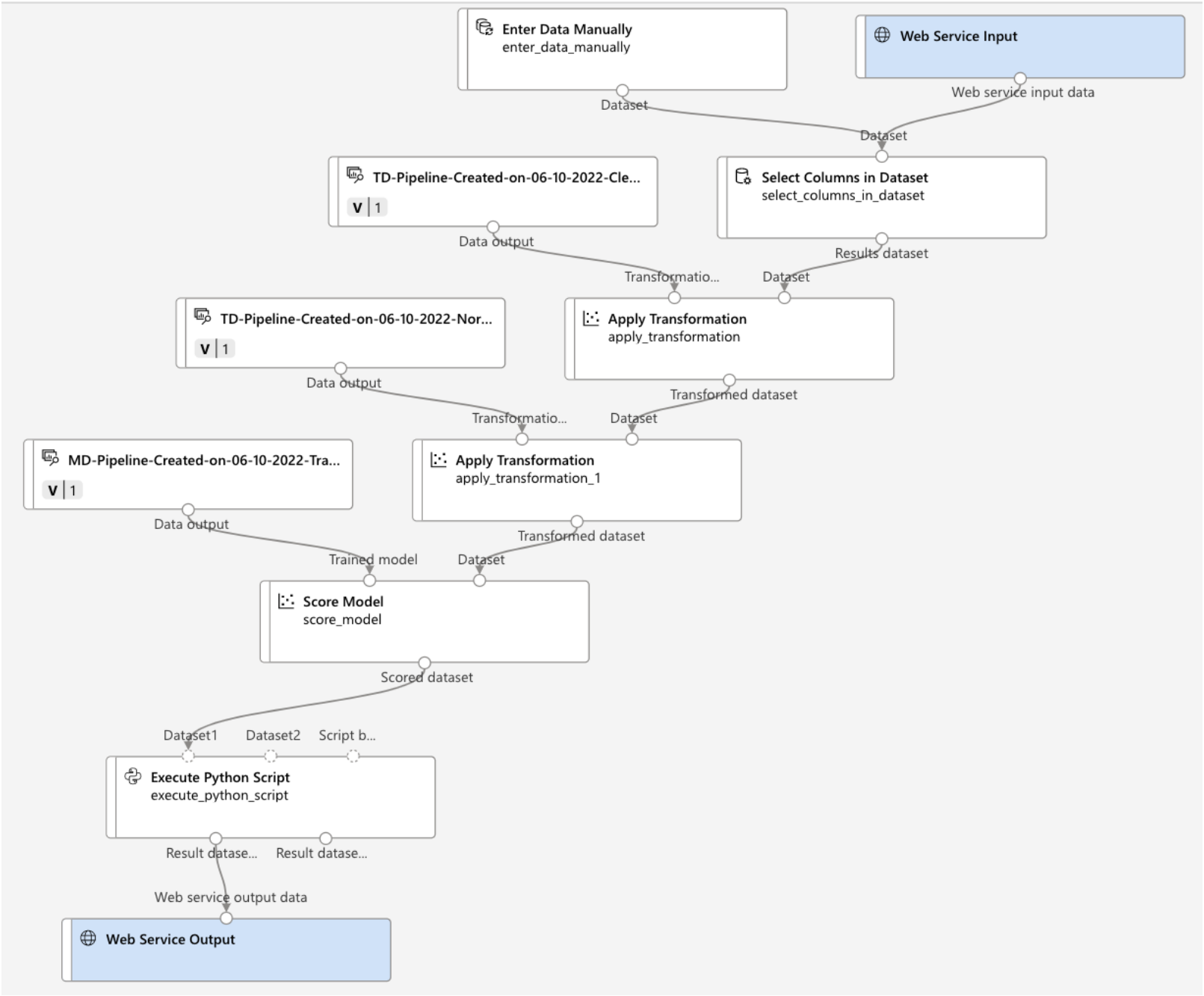
Step 2.5 แก้ python script สำหรับ Execute Python Script

```
import pandas as pd

def azureml_main(dataframe1 = None, dataframe2 = None):

    scored_results = dataframe1[['Scored Labels']]
    scored_results.rename(columns={'Scored Labels':'predicted_price'},
                          inplace=True)
    return scored_results
```

Step 3: Check and run your Inference pipeline

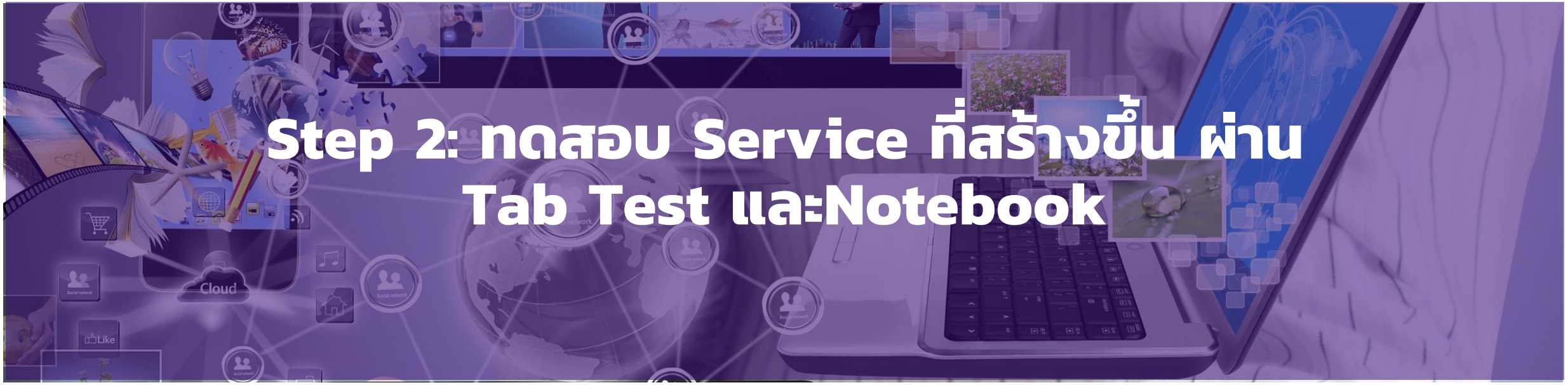


Part

3

Deploy the Inference Pipeline as a Service for apps to use

Step 1: Deploy Real-time Endpoint สำหรับ Predictive Service (ใช้เวลา)



Step 2: ทดสอบ Service ที่สร้างขึ้น ผ่าน Tab Test และ:Notebook

Step 2.1:
Code สำหรับ ทดสอบ Notebook

```
endpoint = 'YOUR_ENDPOINT' #Replace with your endpoint
key = 'YOUR_KEY' #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
      },
    ],
  },
  "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)
  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')))
```

Step 2.2: Copy REST endpoint และ Primary Key ของ Service ที่ deploy สำเร็จ ใส่ใน Notebook

---

Step 2.3: Run Notebook และดูผลลัพธ์ที่ predict ในตัวแปล output: predicted price


Clean up หลังจากทดลอง
เสร็จ โดยการ Delete
Endpoint ที่สร้างขึ้น


End of Lab2

Thank you