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

3

สร้าง Classification Model ด้วย Azure Machine Learning Designer

Classification
(การจัดกลุ่ม / การแยกประเภท)

Classification is a form of machine learning that is used to predict which category, or *class*, an item belongs to.

https://docs.microsoft.com/en-us/learn/modules/create-classification-model-azure-machine-learning-designer/introduction

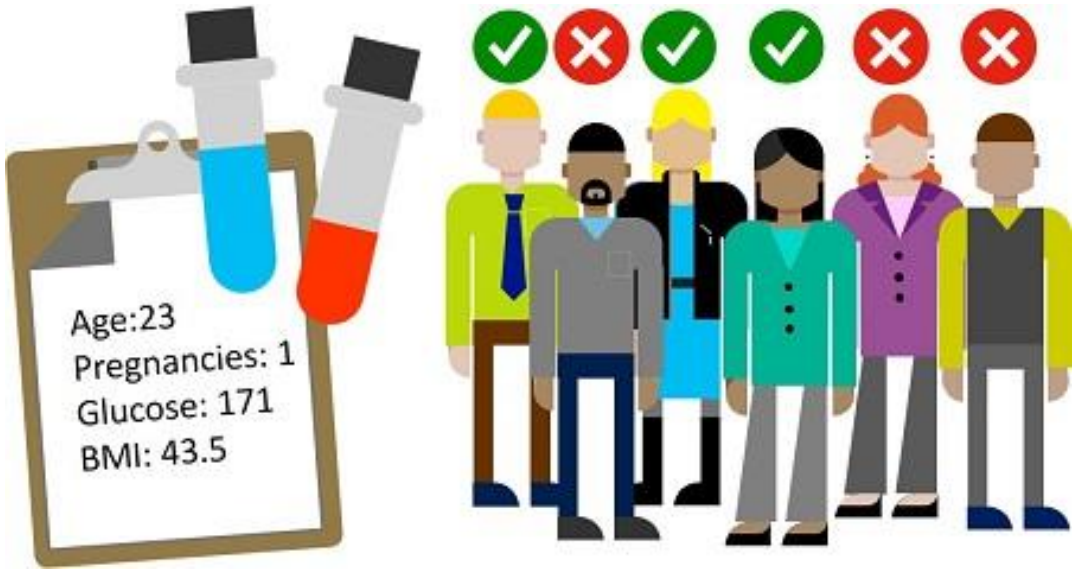
Lab3: สร้าง Classification Model ด้วย Azure Machine Learning Designer

เป้าหมาย:

- ใช้ Azure ML designer สร้าง classification models
  - สร้างและ Run Training Pipeline
  - สร้าง Inference Pipeline
  - Deploy Predictive Model

- Tool: Azure ML Designer
- Model: Two-Class Logistic Regression
- Metric:
  - Accuracy, Precision, Recall, F1 score
  - Confusion Matrix

Case Study และ ข้อมูลที่ใช้



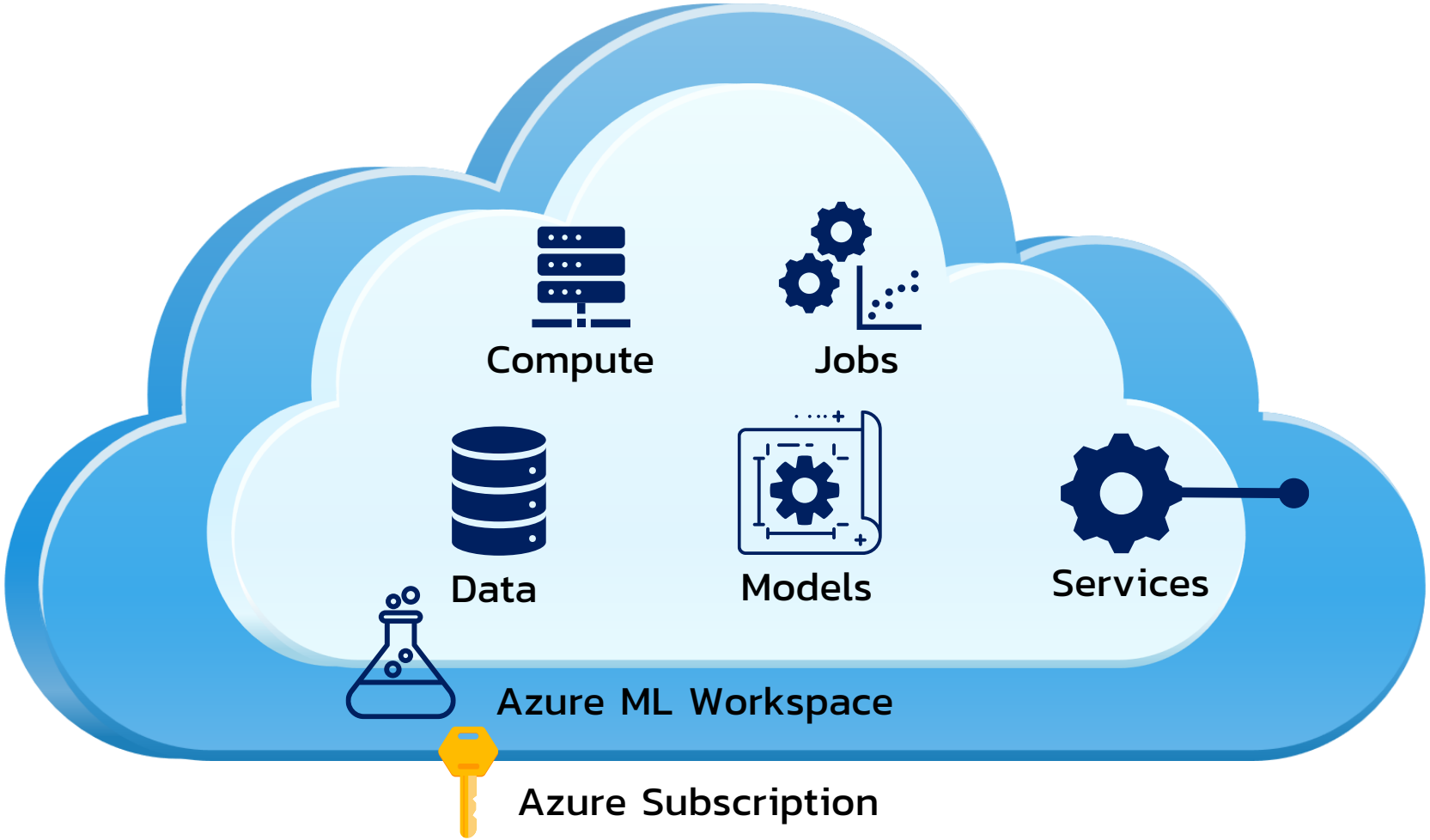
Dataset: ข้อมูลผู้ป่วยเบาหวาน (https://aka.ms/diabetes-data)

Goal:

- เพื่อพยากรณ์ความเสี่ยงในการเป็นโรคเบาหวาน (diabetes) ของผู้ป่วยโดยใช้ข้อมูลผู้ป่วย ได้แก่ age, weight, blood pressure ฯลฯ
- Label ของการจัดกลุ่ม คือ ค่า 0 หรือ 1 ซึ่งหมายถึง ไม่เป็นเบาหวาน (non-diabetic) หรือ or เป็นเบาหวาน (diabetic)

Review: Azure Machine Learning

A cloud-based platform for machine learning



Machine learning in Microsoft Azure

Azure Machine Learning provides the following features and capabilities:

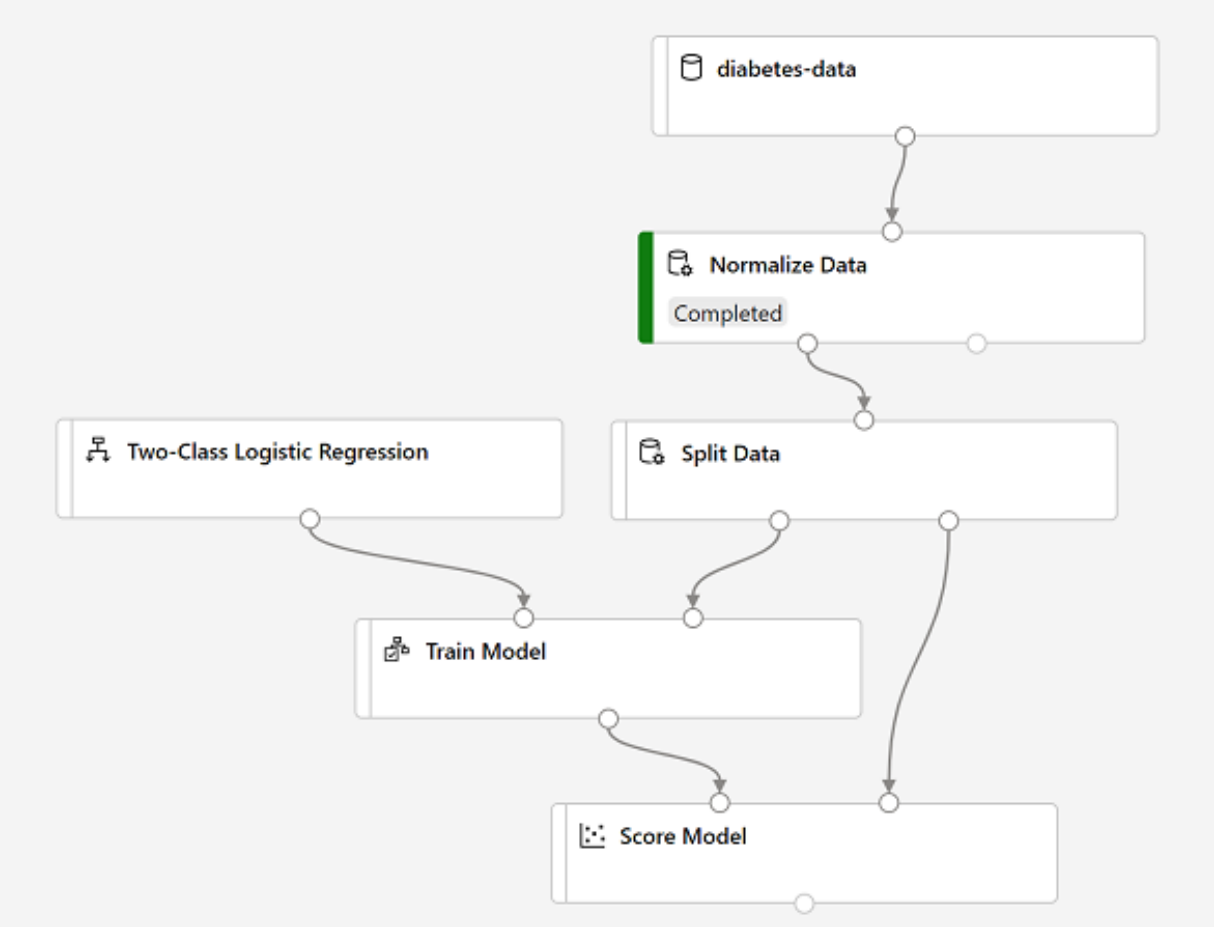
Feature	Capability
Automated machine learning	This feature enables non-experts to quickly create an effective machine learning model from data.
Azure Machine Learning designer	A graphical interface enabling no-code development of machine learning solutions.
Data and compute management	Cloud-based data 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



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 Dataset

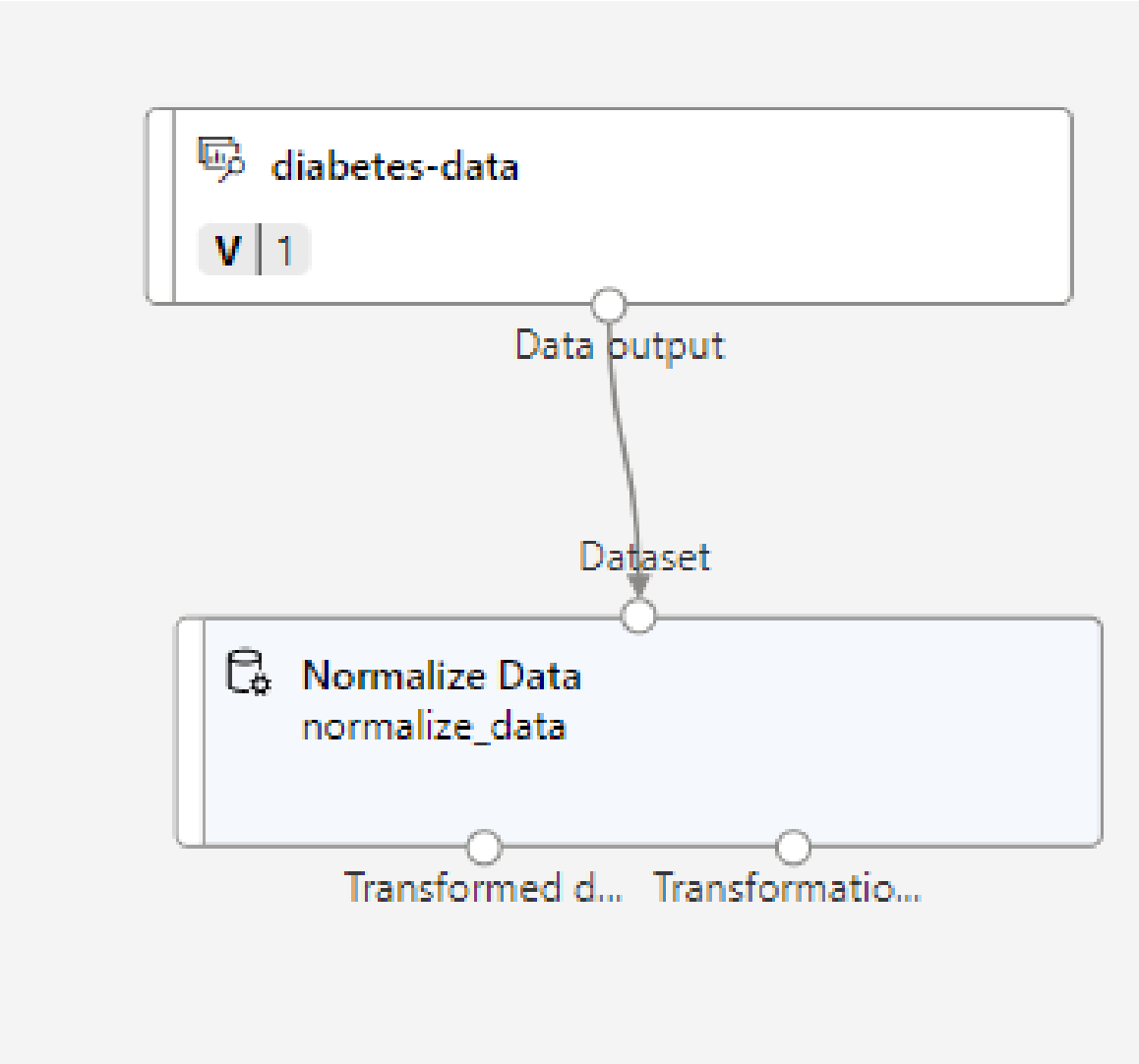
Web URL: https://aka.ms/diabetes-data

Name: diabetes-data

Step 4: สร้าง Pipeline สำหรับ manage dataset

Step 5: ทำ Data Transformation เพื่อจัดการข้อมูลให้พร้อมสำหรับการสร้างโมเดล

- ทำความเข้าใจและสำรวจข้อมูล
- Normalize ข้อมูล ด้วยวิธี MinMax สำหรับ features ต่อไปนี้
  - Pregnancies
  - PlasmaGlucose
  - DiastolicBloodPressure
  - TricepsThickness
  - SerumInsulin
  - BMI
  - DiabetesPedigree
  - Age



Step 6: Run the Pipeline และ explore ผลลัพธ์

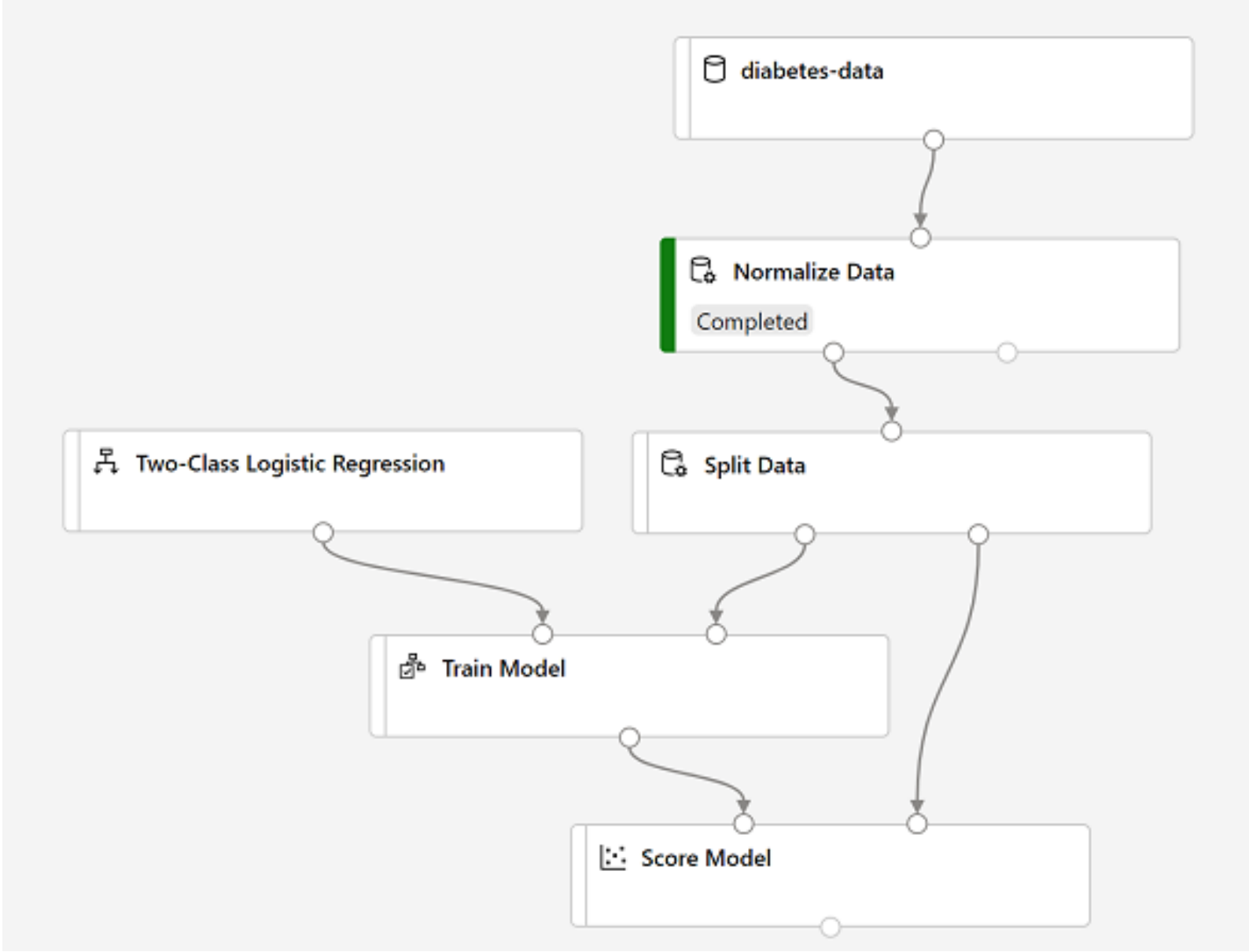
Dataset ได้ถูก Prepare และพร้อมสำหรับการสร้าง Model

Transformed_dataset

Rows 10,000 Columns 10

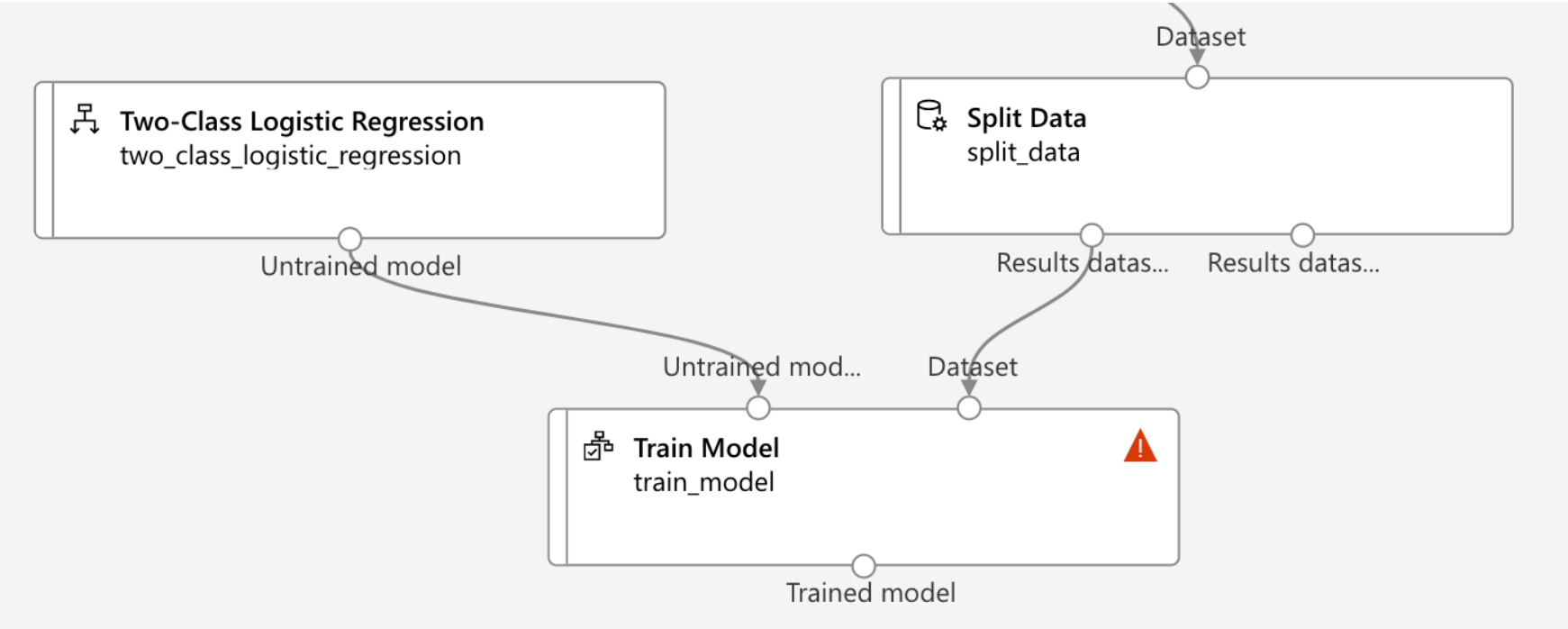
s	PlasmaGlucose	DiastolicBloodPressure	TricepsThickness	SerumInsulin	BMI	DiabetesPedigree	Age
	0.858108	0.602151	0.317647	0.011509	0.66895	0.510511	0
	0.324324	0.741935	0.470588	0.028133	0.080345	0.036123	0.035714
	0.47973	0.247312	0.529412	0.026854	0.616134	0.000438	0.035714
	0.398649	0.580645	0.211765	0.370844	0.300826	0.541848	0.392857
	0.277027	0.376344	0.235294	0.026854	0.645024	0.212047	0.017857
	0.256757	0.731183	0.023529	0.305627	0.040264	0.011414	0.089286
	0.601351	0.247312	0.141176	0.272379	0.098868	0.043226	0
	0.155405	0.677419	0.423529	0.028133	0.002033	0.071112	0.089286

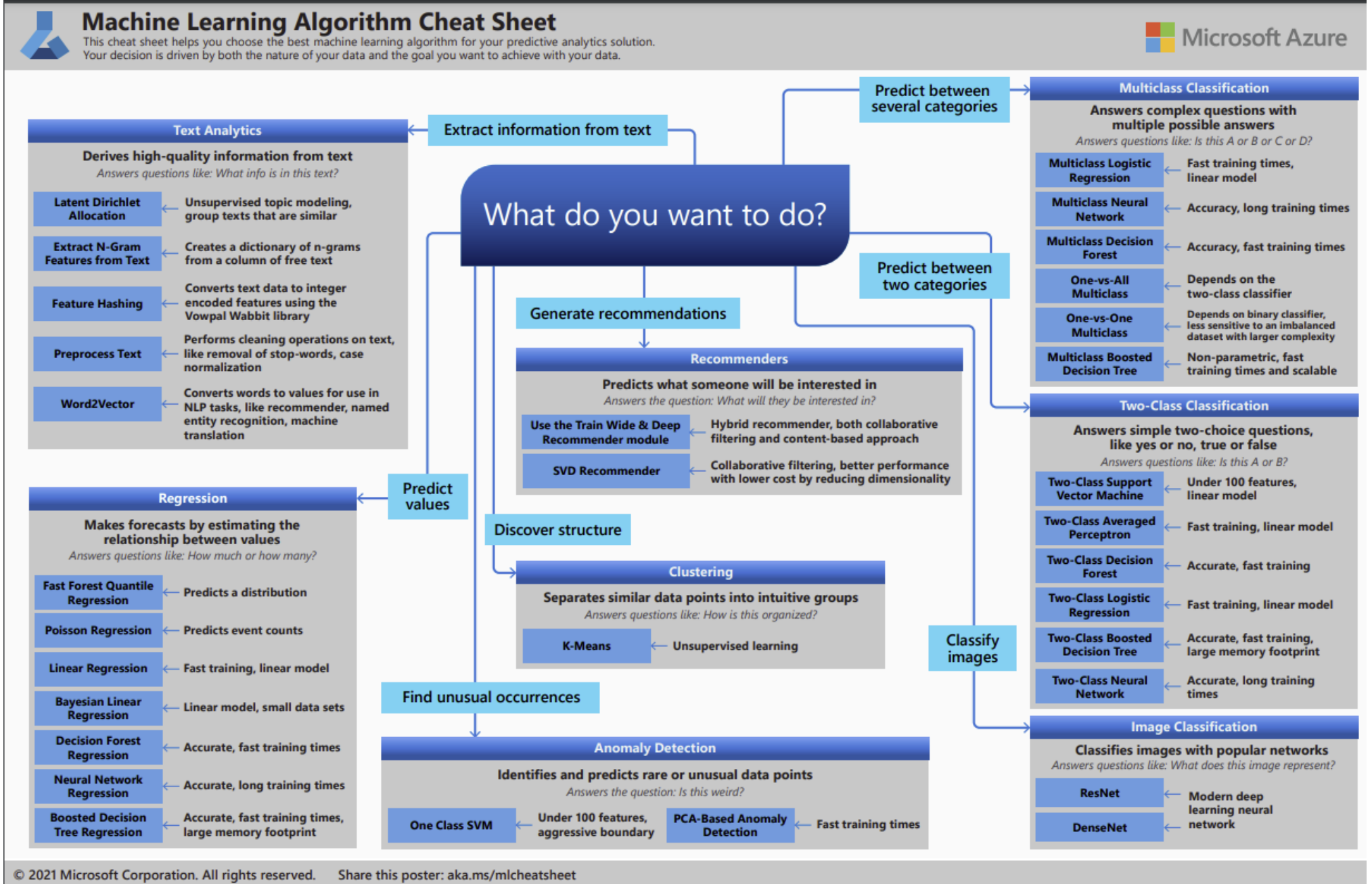
Step 7: Create Training Pipeline for Model Training



Using a *classification* algorithm

- The **Diabetic** label the model will predict is a class (0 or 1).
- So, there are two possible classes, so we need a *binary classification* algorithm.
- Connect **Two-Class Logistic Regression** module
  - Connect **Two-Class Logistic Regression** output to the **Untrained model** (left) input of the **Train Model** module
- See more algorithms: https://docs.microsoft.com/en-us/azure/machine-learning/algorithm-cheat-sheet

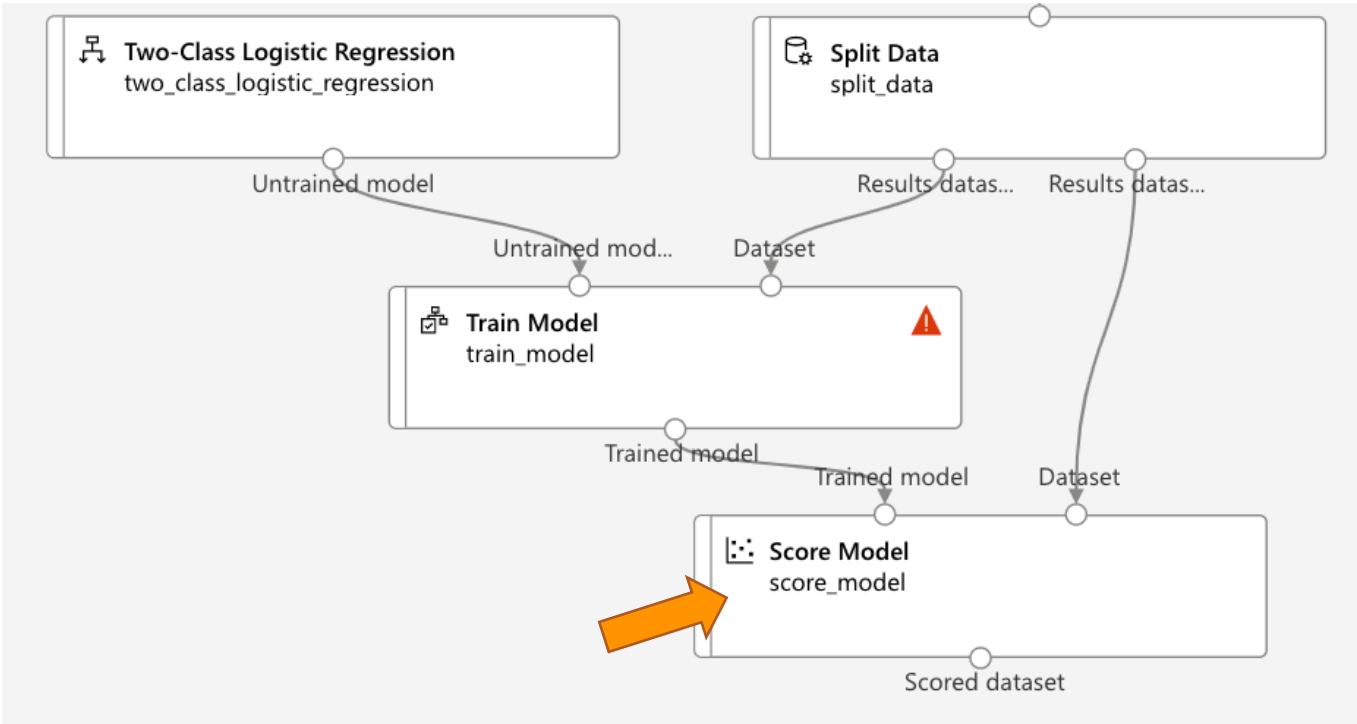




Source: https://docs.microsoft.com/en-us/azure/machine-learning/algorithm-cheat-sheet

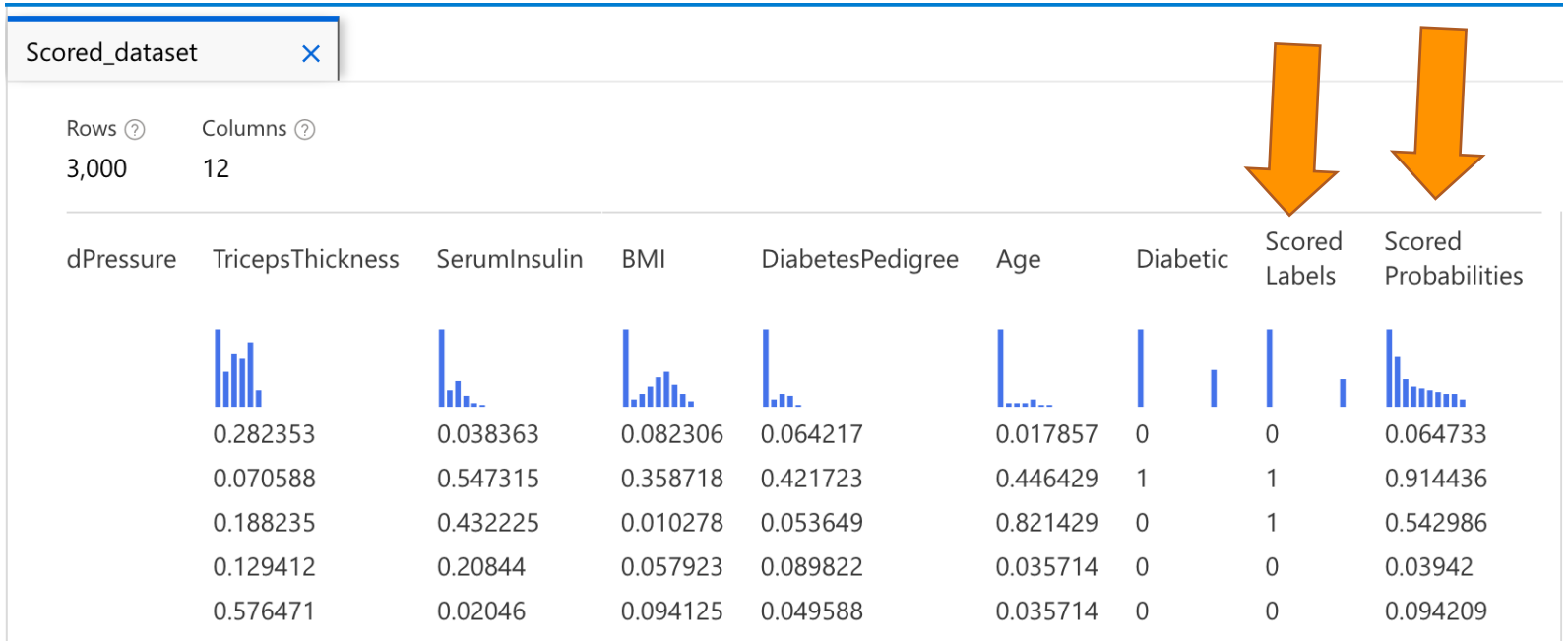
Score Model

- To test the trained model, use the trained model to **Score** (Predict the price) the validation dataset Connect **Score Model** module
- Connect the trained model and the validation dataset to **Score Model**
- **Run** the training pipeline (Click Submit)



See Result

- When it has completed, Click Job detail
- select the **Score Model** module
- on the **Outputs + logs** tab, under **Data outputs**, select the **Preview Data** icon to view the results.
- **Scored labels:** the predicted label values.
- **Scored Probabilities:** a probability value between 0 and 1.
  - The probabilities greater than 0.5 result in a predicted label of ***1*** (diabetic)



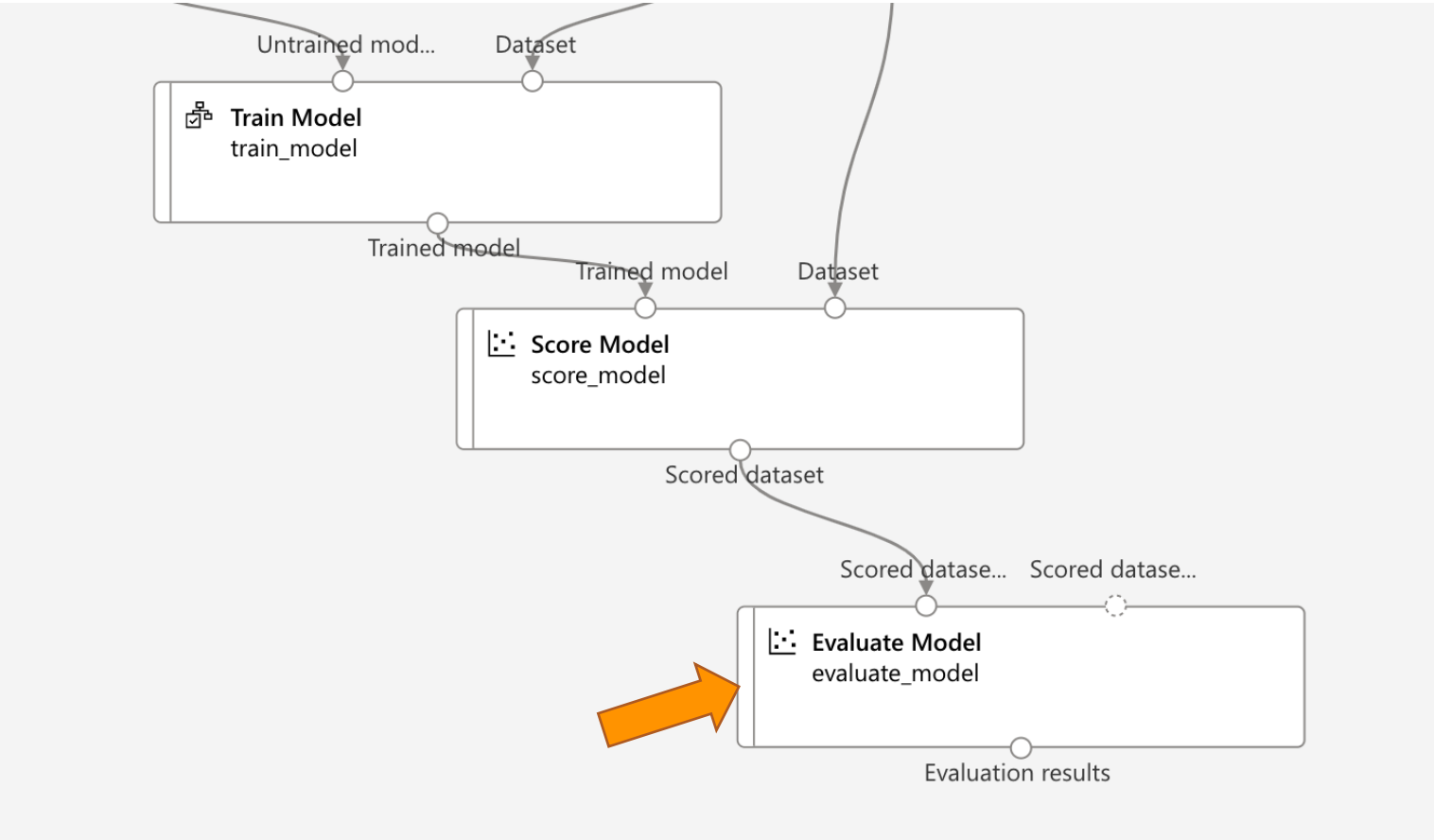
Scored_dataset

Rows 3,000 Columns 12

dPressure	TricepsThickness	SerumInsulin	BMI	DiabetesPedigree	Age	Diabetic	Scored Labels	Scored Probabilities
	0.282353	0.038363	0.082306	0.064217	0.017857	0	0	0.064733
	0.070588	0.547315	0.358718	0.421723	0.446429	1	1	0.914436
	0.188235	0.432225	0.010278	0.053649	0.821429	0	1	0.542986
	0.129412	0.20844	0.057923	0.089822	0.035714	0	0	0.03942
	0.576471	0.02046	0.094125	0.049588	0.035714	0	0	0.094209

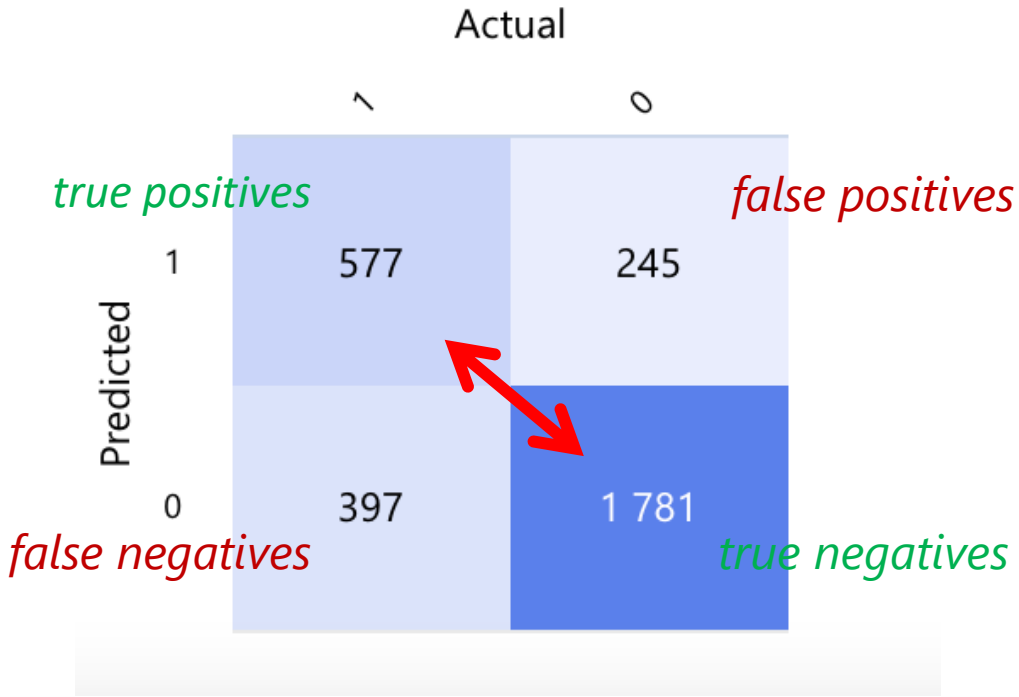
Step 8: Evaluate the Model

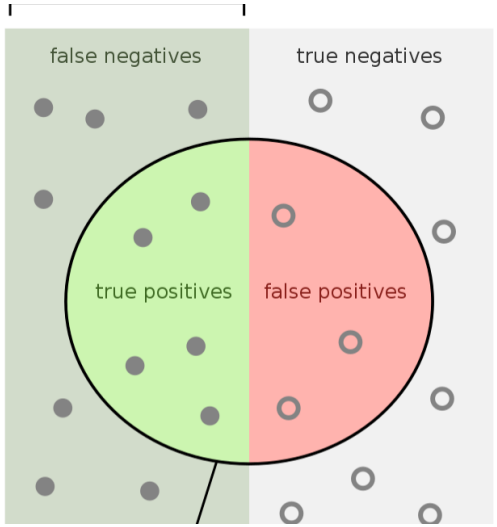
- Use the **Evaluate Model** module to evaluate the performance of the model.
- Connect the **Evaluate Model** module.
- **Run**



Confusion Matrix

- A tabulation of the predicted and actual value counts for each possible class.
- For a binary classification model, where you're predicting one of 2 possible values, the confusion matrix is a 2x2 grid showing the predicted and actual value counts for classes **0** and **1**
- See a diagonal line of cells for the predicted and actual labels match.





How many retrieved items are relevant?

Precision =

How many relevant items are retrieved?

Recall =

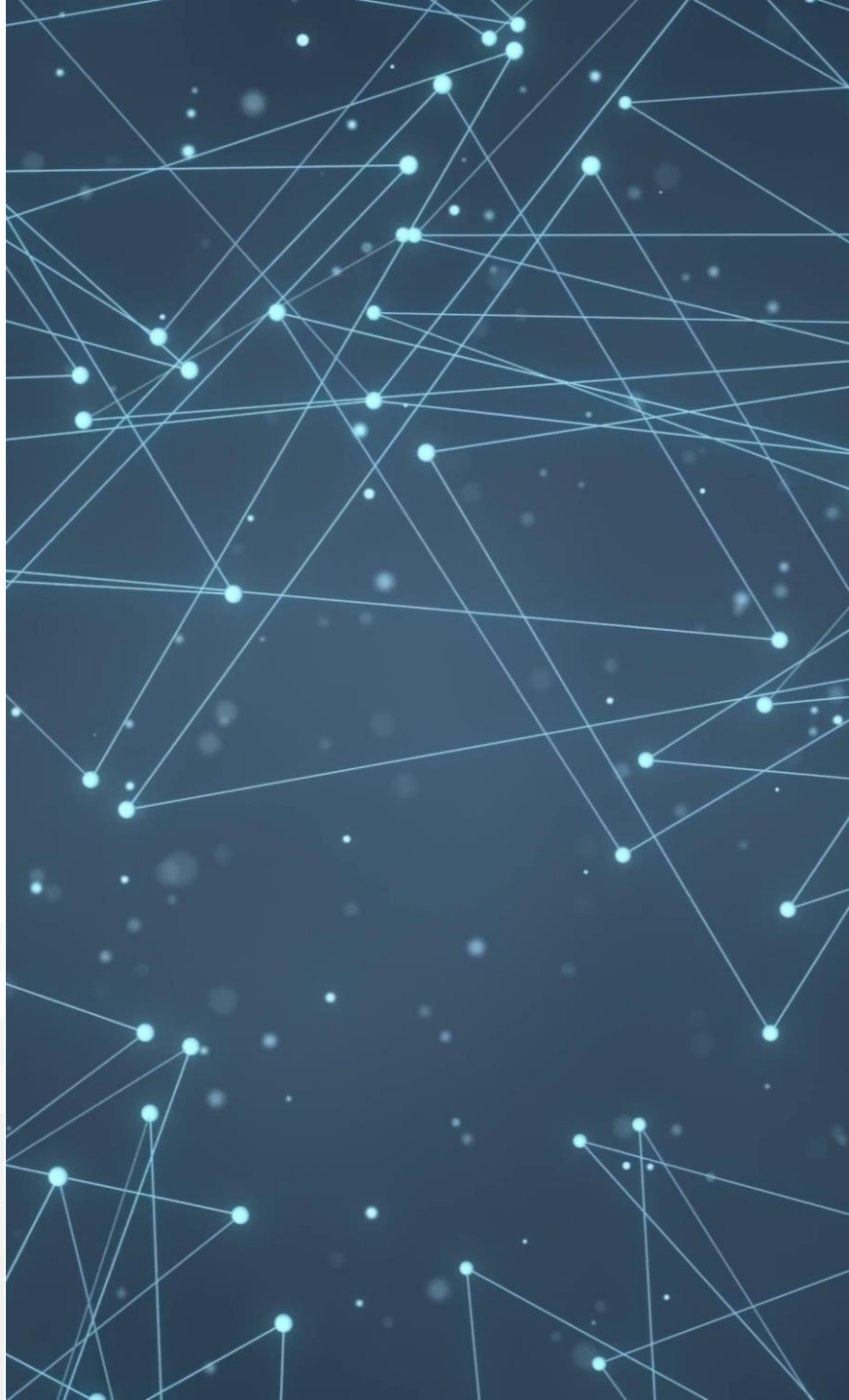
https://en.wikipedia.org/wiki/Precision_and_recall

Other Evaluation Metrics

Metric	Description
Accuracy	The ratio of correct predictions (True positives + True negatives) to the total number of predictions. In other words, what proportion of diabetes predictions did the model get right?
Precision	The fraction of positive cases correctly identified (The number of true positives divided by the number of true positives plus false positives). In other words, out of all the patients that the model predicted as having diabetes, how many are actually diabetic?
Recall	The fraction of the cases classified as positive that are actually positive (The number of true positives divided by the number of true positives plus false negatives). In other words, out of all the patients who actually have diabetes, how many did the model identify?
F1 Score	An overall metric that essentially combines precision and recall.

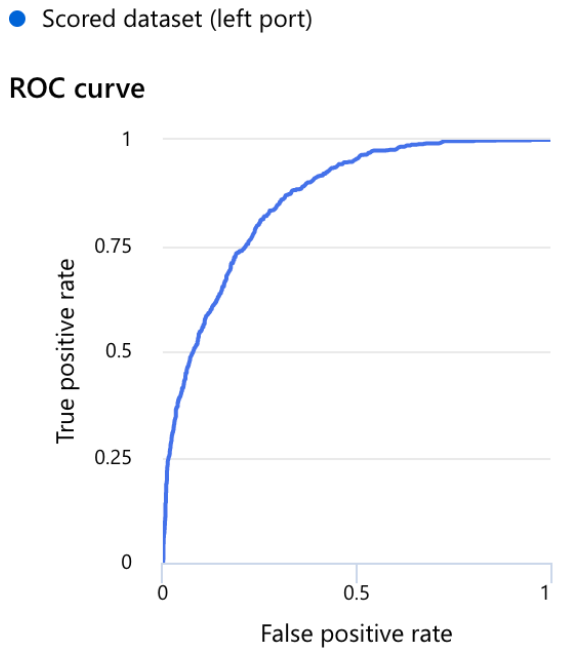
Threshold Slider

- Remember that what a classification model predicts is the probability for each possible class.
- prediction is a value between 0 and 1.
- By default, a predicted probability for diabetes *including or above* 0.5 results in a class prediction of 1,
- And *below* 0.5 means that there's a greater probability of the patient **not** having diabetes.
- Try moving the threshold slider and observe the effect on the confusion matrix.



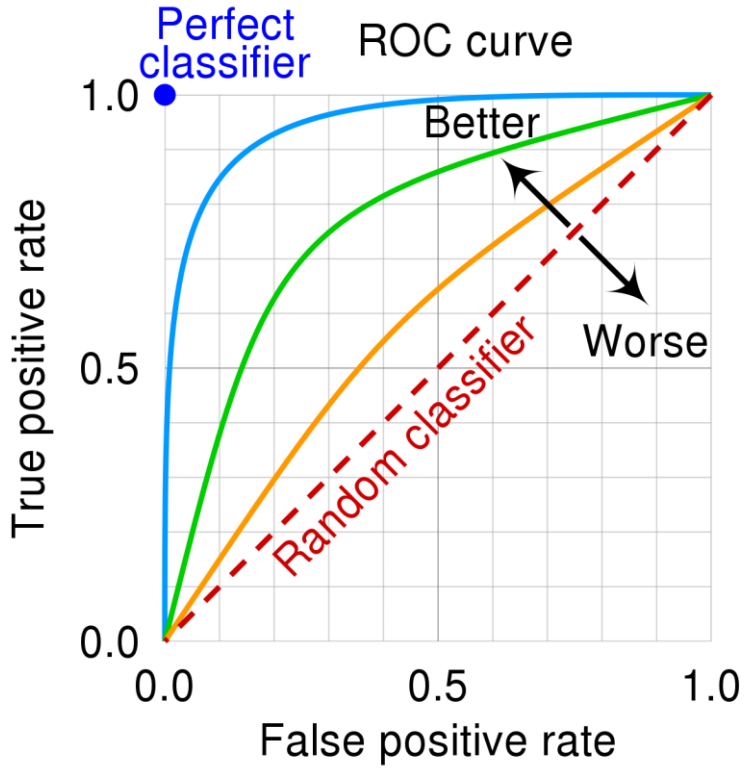
ROC Curve

Receiver Operating Characteristic



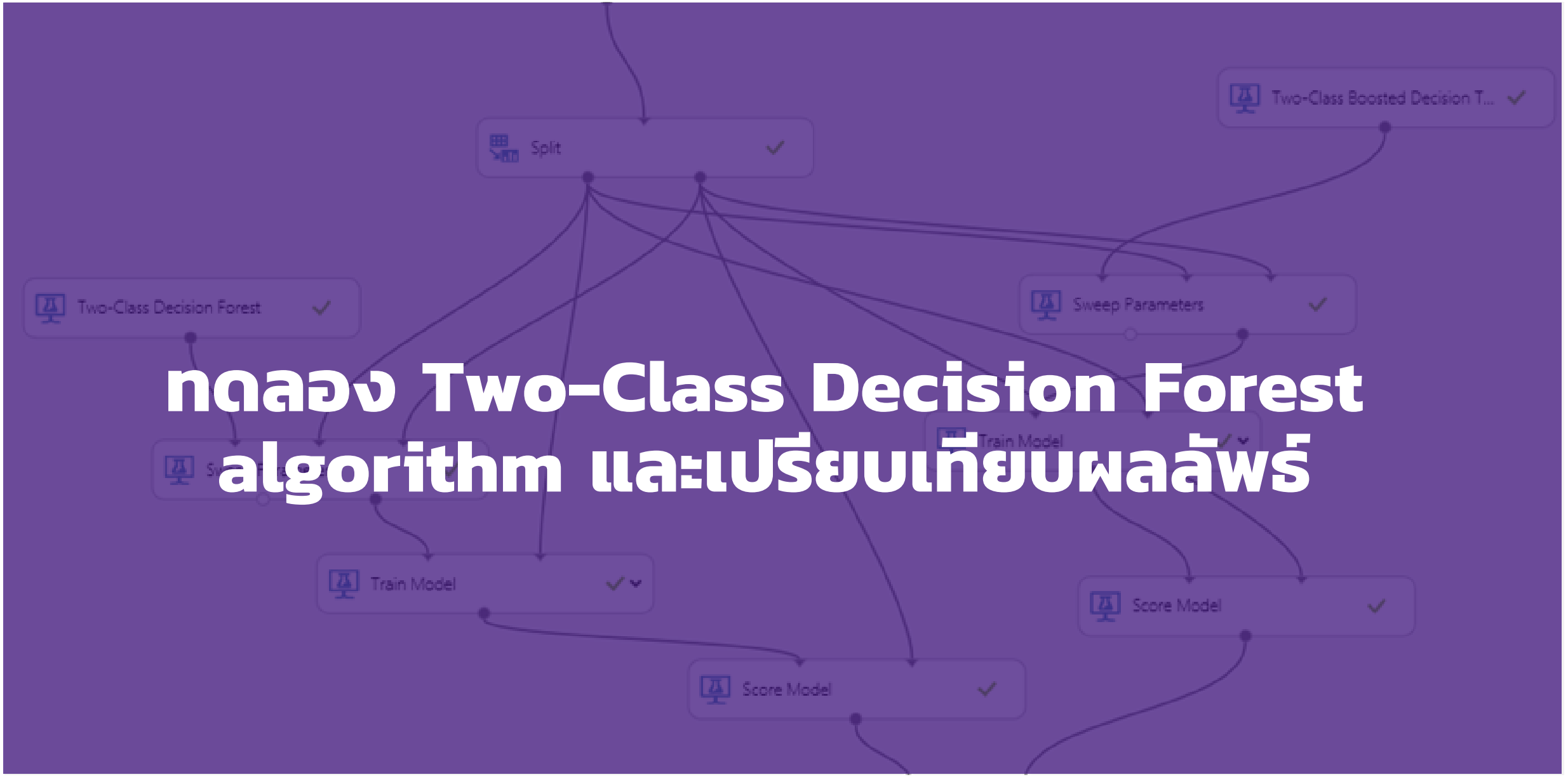
The area under the diagonal line represents an **AUC of 0.5**. If the AUC for your model is **higher than this** for a binary classification model, then the model performs **better than a random guess**.

The ROC space for a "better" and "worse" classifier.



https://en.wikipedia.org/wiki/Receiver_operating_characteristic

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

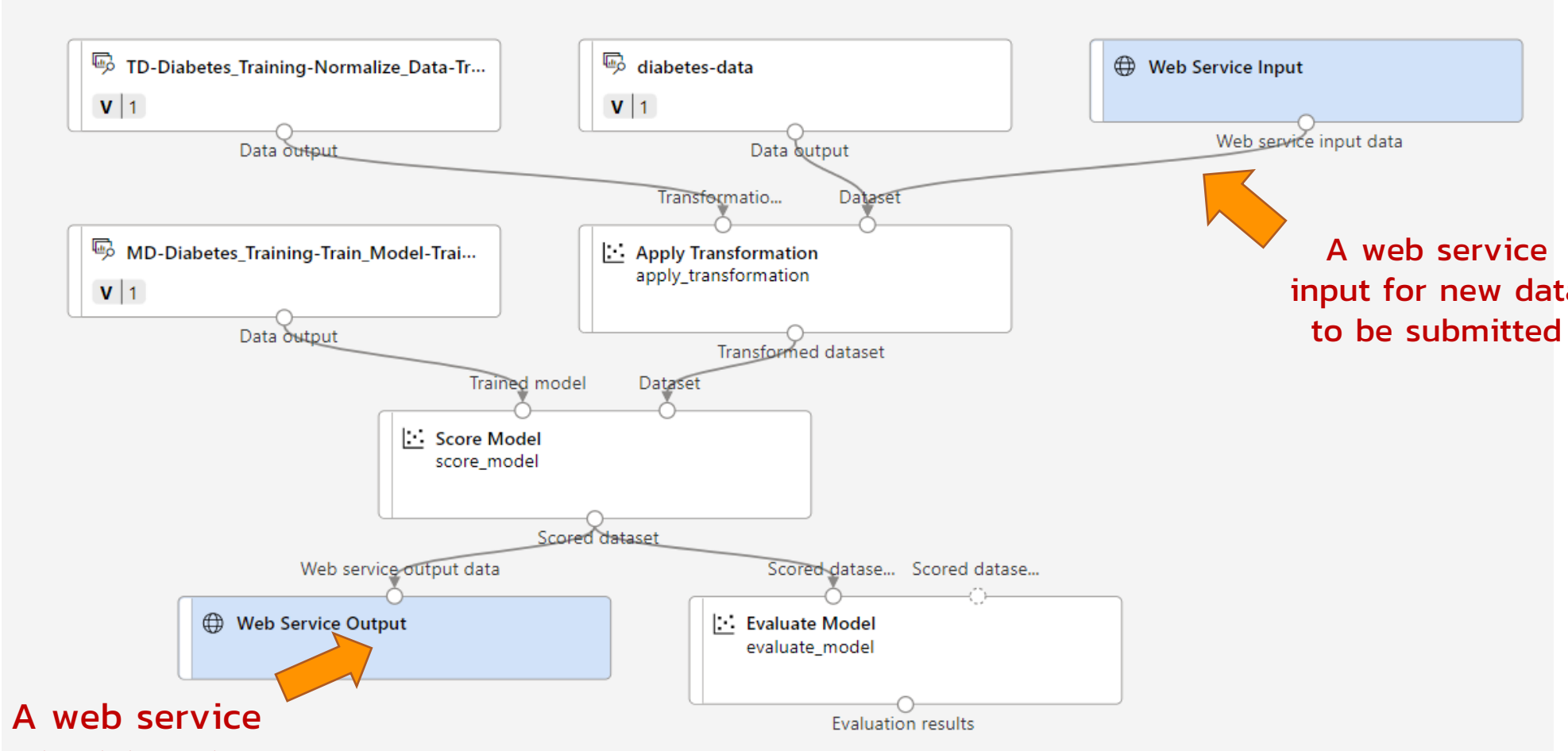


ทดลอง Two-Class Decision Forest algorithm และเปรียบเทียบผลลัพธ์

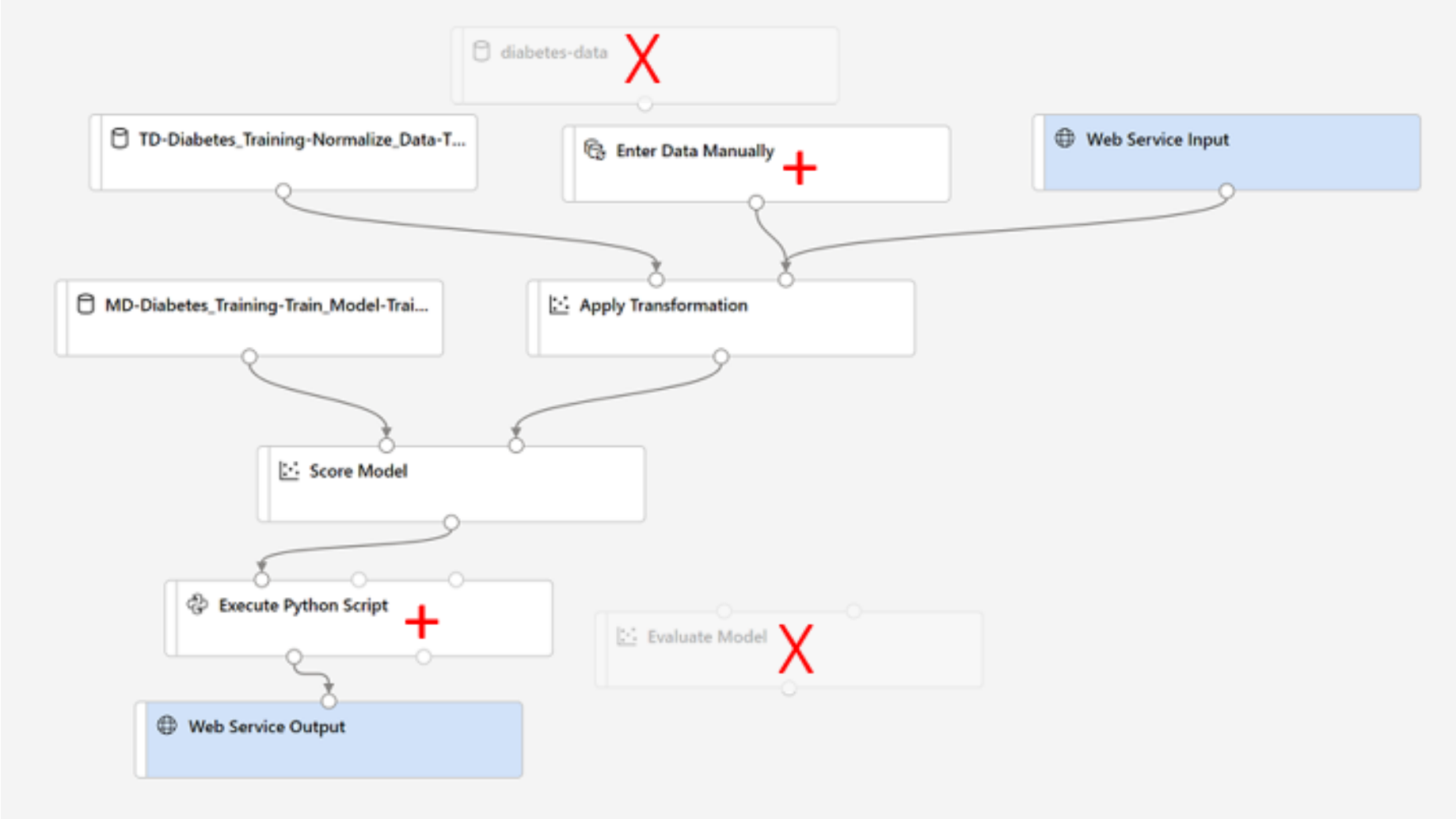
Part 2

Create an Inference Pipeline to predict labels from new data

Step 1: สร้าง Inference Pipeline



Step 2: แก้ไข the Inference Pipeline



Step 2.1 ลบ Diabetes data

Step 2.2 เพิ่ม Enter Data Manually โดย copy&paste ข้อมูล CSV data ของ 3 patient ดังนี้

PatientID,Pregnancies,PlasmaGlucose,DiastolicBloodPressure,TricepsThickness,SerumInsulin,BMI,DiabetesPedigree,Age
1882185,9,104,51,7,24,27.36983156,1.350472047,43
1662484,6,73,61,35,24,18.74367404,1.074147566,75
1228510,4,115,50,29,243,34.69215364,0.741159926,59

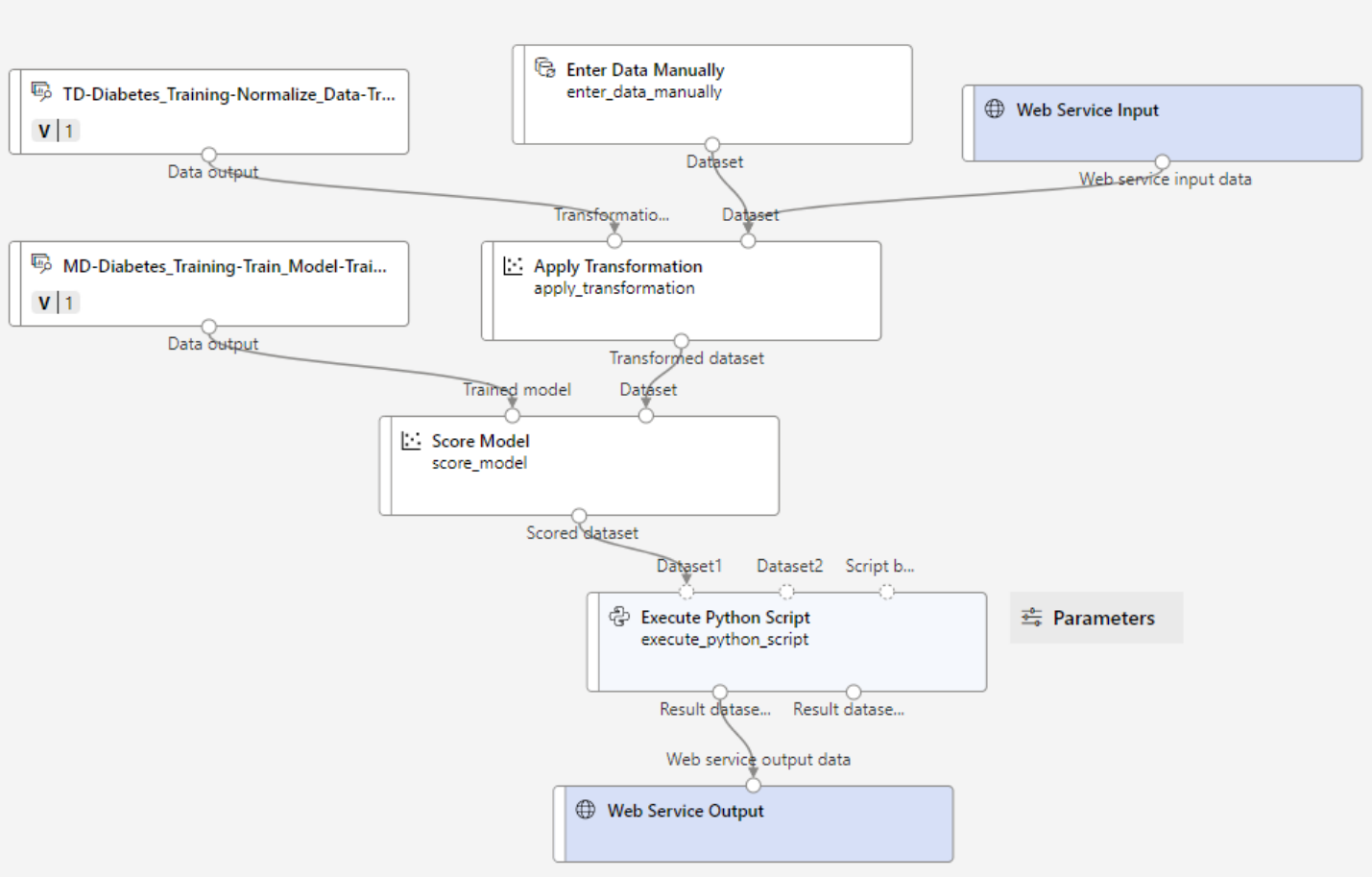
Step 2.3 ลบ **Evaluate Model** ออก

Step 2.4 เพิ่ม **Execute Python Script** เพื่อคืนค่าผลลัพธ์ **PatientID**, **Scored Labels** and **Scored Probabilities** 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[['PatientID', 'Scored
Labels', 'Scored Probabilities']]
    scored_results.rename(columns={'Scored
Labels':'DiabetesPrediction',
                                   'Scored
Probabilities':'Probability'},
                          inplace=True)
    return scored_results
```

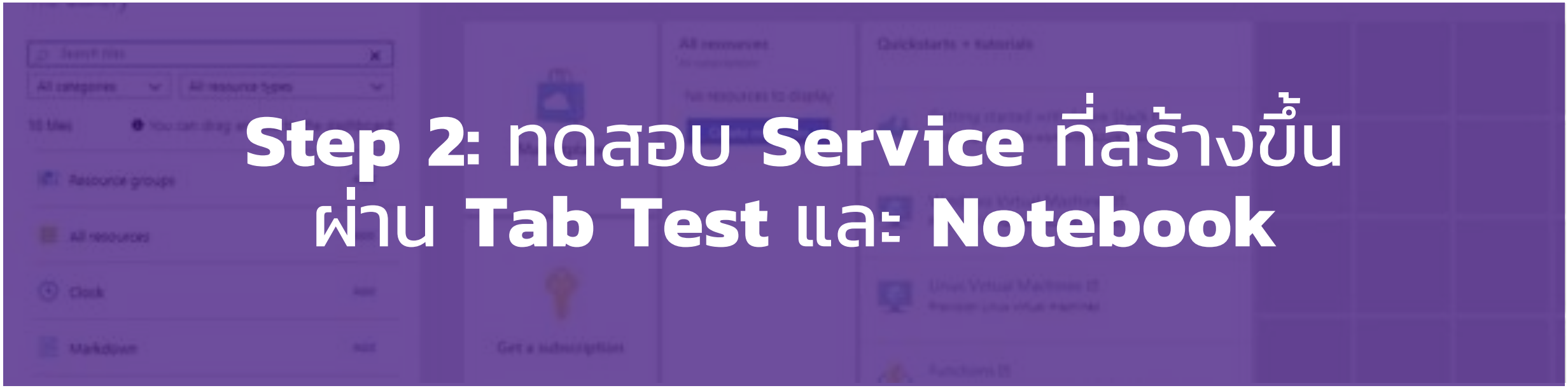
Step 3: Check and run your Inference pipeline



Part 3

Deploy the Inference Pipeline as a Service for apps to use

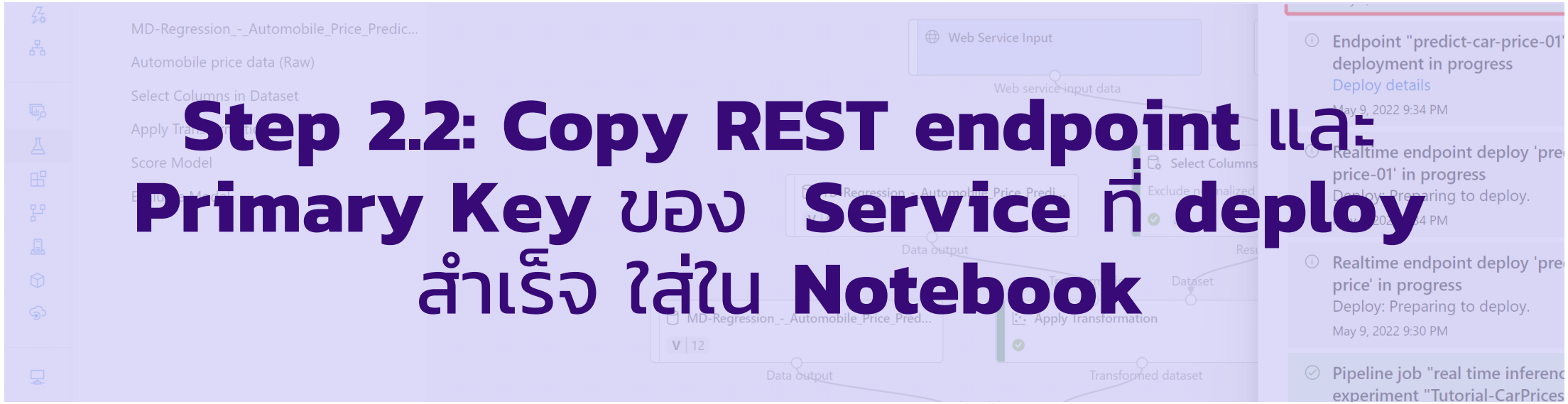
Step 1: Deploy Real-time Endpoint สำหรับ Predictive Service (ใช้เวลา)



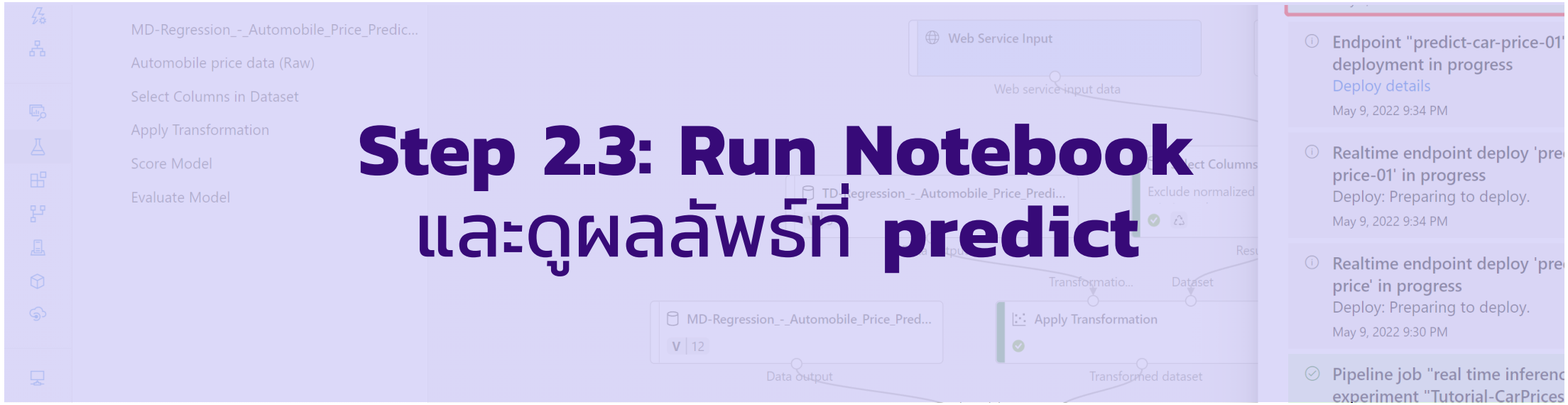
Step 2: ทดสอบ Service ที่สร้างขึ้น ผ่าน Tab Test และ Notebook

Step 2.1: code สำหรับทดสอบ Notebook

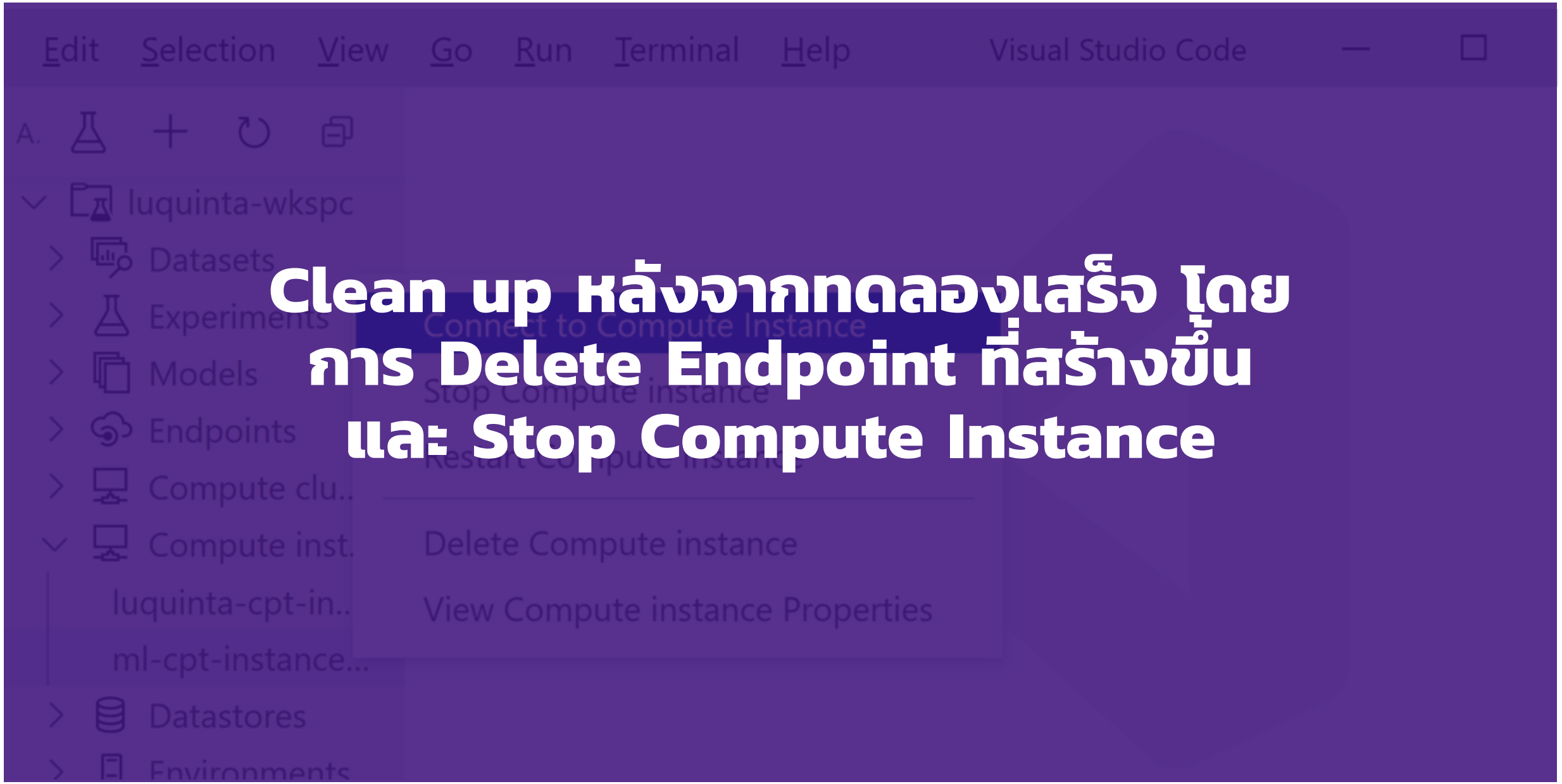
```
endpoint = 'YOUR_ENDPOINT' #Replace with your endpoint
key = 'YOUR_KEY' #Replace with your key

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')))
```



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



Step 2.3: Run Notebook และดูผลลัพธ์ที่ predict



Clean up หลังจากทดลองเสร็จ โดยการ Delete Endpoint ที่สร้างขึ้น และ Stop Compute Instance


End of Lab3

Thank you