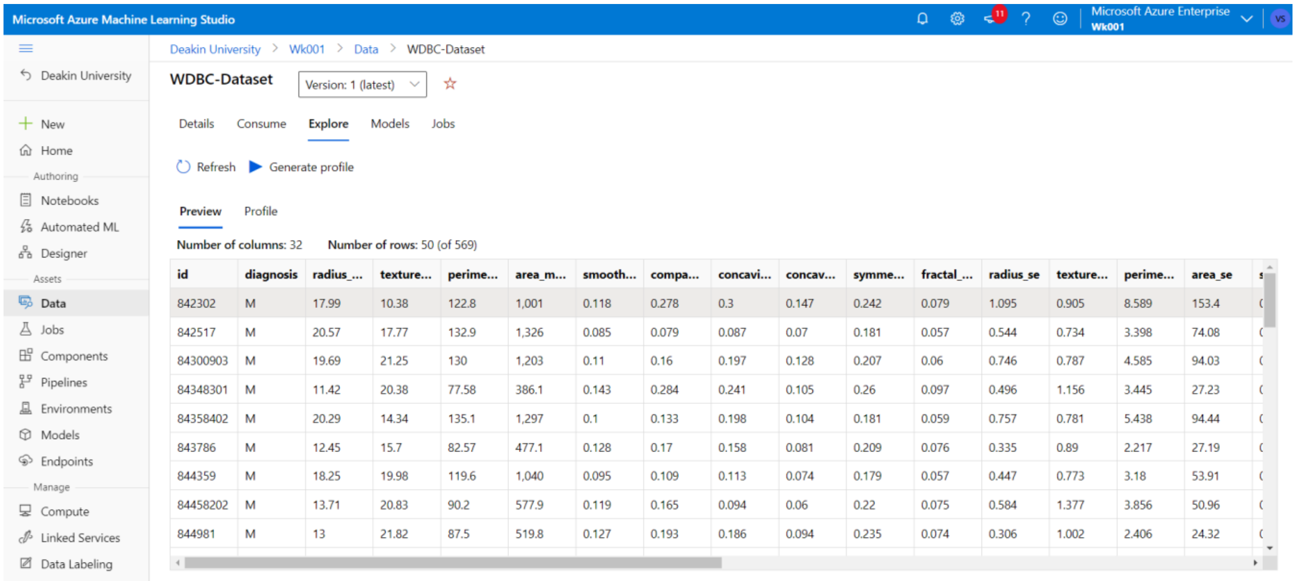
Task 4.1: Design and deploy the model using Azure machine learning

A. Data Ingestion



WDBC-Dataset Version: 1 (latest)

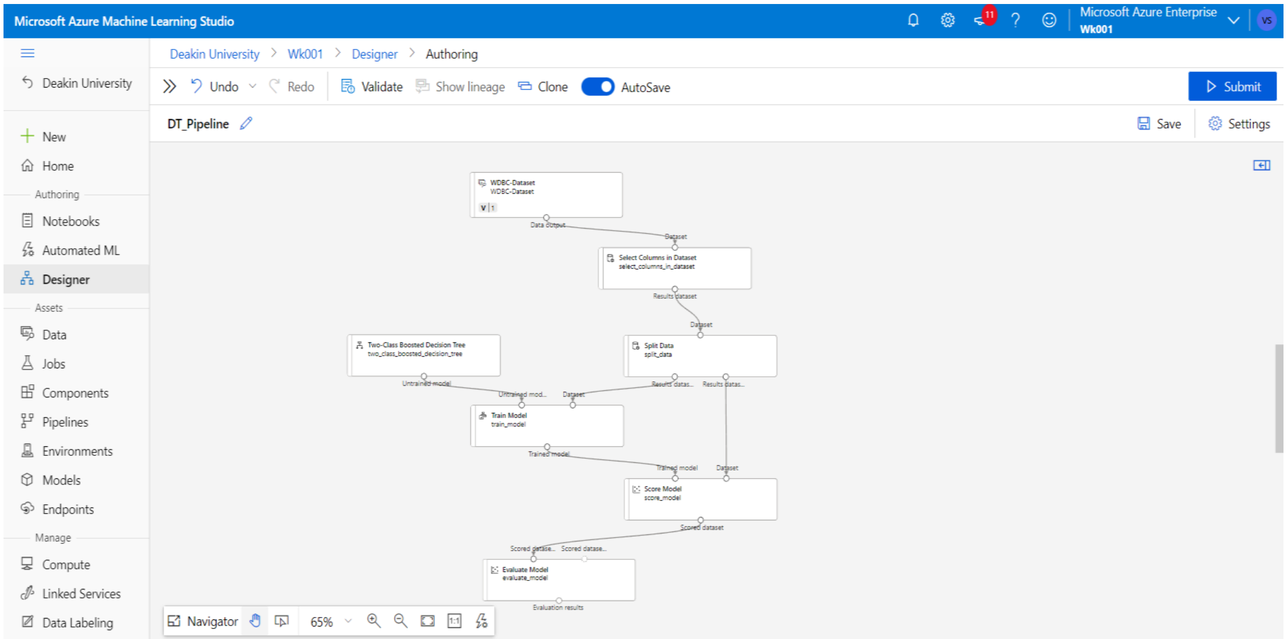
Number of columns: 32 Number of rows: 50 (of 569)

id	diagnosis	radius_...	texture...	perime...	area_m...	smooth...	compa...	concavi...	concav...	symme...	fractal_...	radius_se	texture...	perime...	area_se
842302	M	17.99	10.38	122.8	1,001	0.118	0.278	0.3	0.147	0.242	0.079	1.095	0.905	8.589	153.4
842517	M	20.57	17.77	132.9	1,326	0.085	0.079	0.087	0.07	0.181	0.057	0.544	0.734	3.398	74.08
84300903	M	19.69	21.25	130	1,203	0.11	0.16	0.197	0.128	0.207	0.06	0.746	0.787	4.585	94.03
84348301	M	11.42	20.38	77.58	386.1	0.143	0.284	0.241	0.105	0.26	0.097	0.496	1.156	3.445	27.23
84358402	M	20.29	14.34	135.1	1,297	0.1	0.133	0.198	0.104	0.181	0.059	0.757	0.781	5.438	94.44
843786	M	12.45	15.7	82.57	477.1	0.128	0.17	0.158	0.081	0.209	0.076	0.335	0.89	2.217	27.19
844359	M	18.25	19.98	119.6	1,040	0.095	0.109	0.113	0.074	0.179	0.057	0.447	0.773	3.18	53.91
84458202	M	13.71	20.83	90.2	577.9	0.119	0.165	0.094	0.06	0.22	0.075	0.584	1.377	3.856	50.96
844981	M	13	21.82	87.5	519.8	0.127	0.193	0.186	0.094	0.235	0.074	0.306	1.002	2.406	24.32

The initial step in deploying the model is to create a pipeline using prebuilt components for data ingestion, model building, training, & deployment. To ingest the data, we click on the Data Tab, provide a name, description, & type for a dataset, and then upload the file from our local machine.

In this case, we have loaded the Wisconsin Diagnostic Breast Cancer (WDBC) dataset, which is used to classify breast cancer tumours as malignant or benign with high accuracy (Frank 2022:3211). The dataset contains information on 569 samples, each with 30 different input features. The first column provides a unique identifier for each sample, while the second column indicates the tumour's malignancy status. The remaining 30 columns capture various measurements of cell nuclei, including mean, standard error, and worst values for features such as radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension.

B. Designing the Decision Tree Model in the experiment section

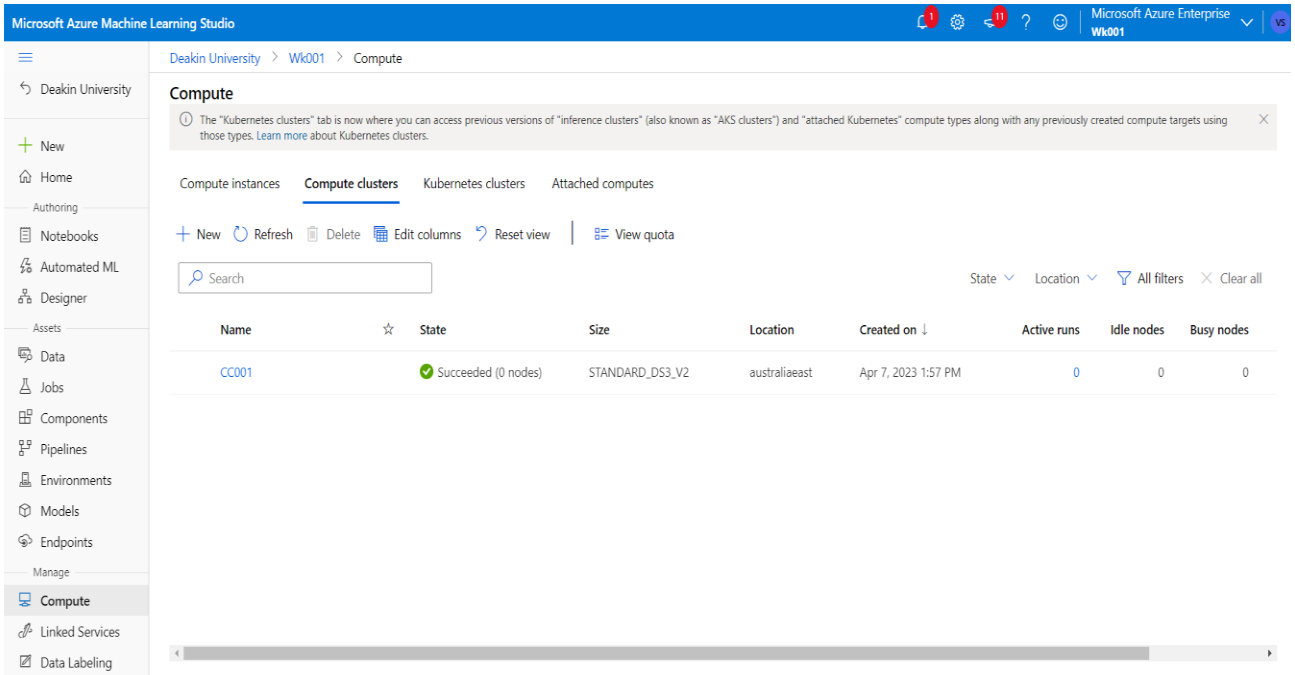


In the Azure experiment section, we can design a decision tree model for breast cancer classification using the following steps:

1. Select the uploaded WDBC dataset and choose it as the dataset for the experiment.

2. Perform feature selection to only use the relevant columns for training the model. In this case, exclude the "Id" column as it is not important for model training.
3. Use the prebuilt "Split Data" component to split the dataset into 75% for training and 25% for testing.
4. Choose the decision tree algorithm from the available prebuilt components in Azure and connect it to the data for model training.
5. Initiate the training process by using the "Train Model" component for the decision tree model.
6. Use the "Score Model" component to apply the trained model to new data and generate predictions or outputs based on the patterns learned from the model.
7. Once the training is complete, evaluate the performance of the decision tree model using the "Evaluate Model" component, which displays metrics such as accuracy, precision, recall, and F1-score (George 2022).

C. Create a Compute Cluster for Model Training



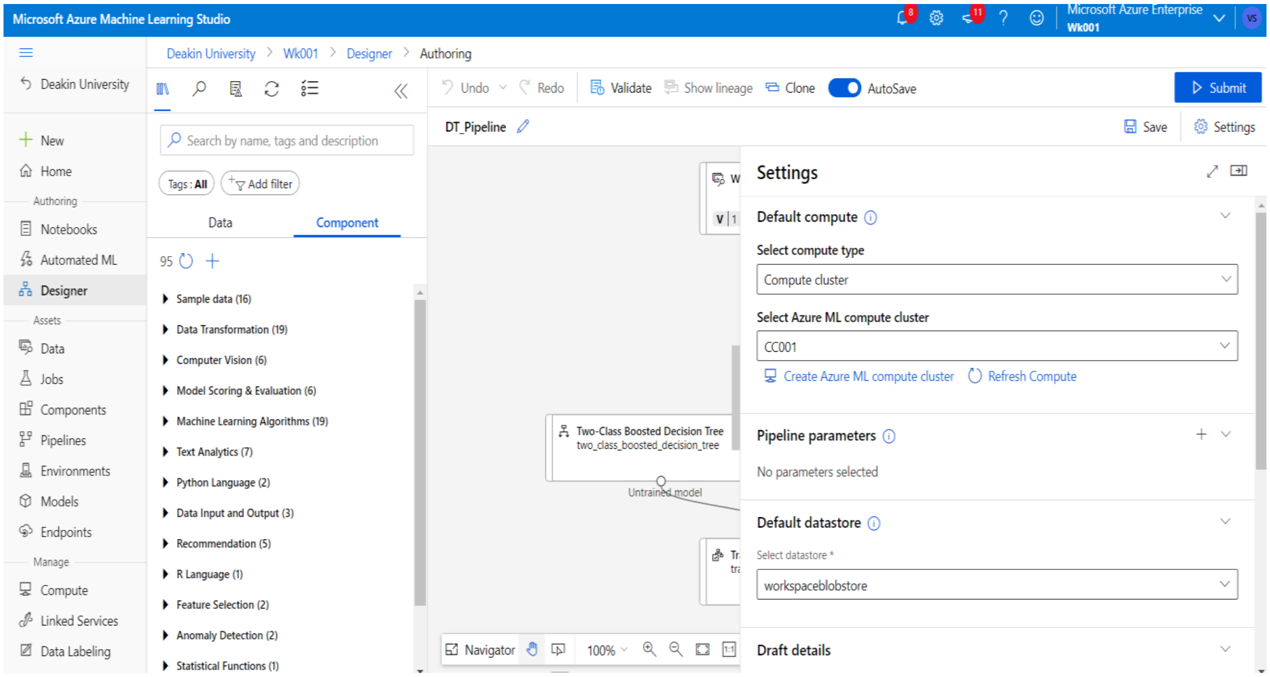
Compute

The "Kubernetes clusters" tab is now where you can access previous versions of "inference clusters" (also known as "AKS clusters") and "attached Kubernetes" compute types along with any previously created compute targets using those types. Learn more about Kubernetes clusters.

Name	State	Size	Location	Created on ↓	Active runs	Idle nodes	Busy nodes
CC001	✓ Succeeded (0 nodes)	STANDARD_DS3_V2	australiaeast	Apr 7, 2023 1:57 PM	0	0	0

As part of our deployment process, we have established a compute cluster named "CC001" in Azure to create a dedicated computing environment for training machine learning models. This involves selecting the location and creating a cluster of virtual machines (VMs) or other optimized computing resources specifically designed for machine learning tasks like model training and data processing.

D. Train the pipeline model



Settings

Default compute

Select compute type: Compute cluster

Select Azure ML compute cluster: CC001

Pipeline parameters

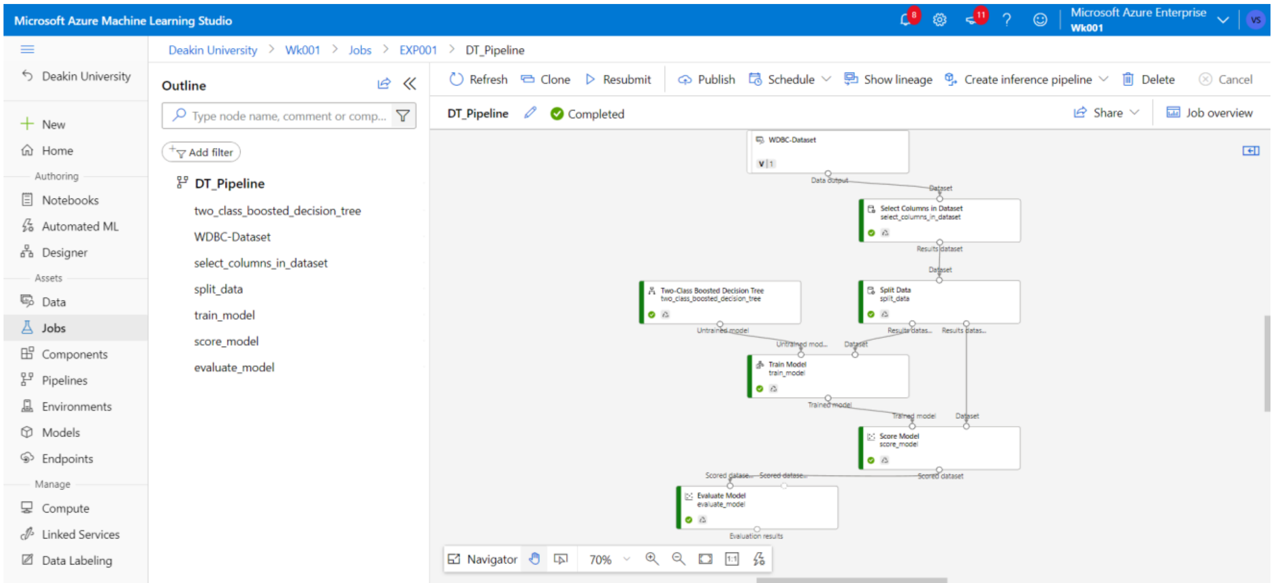
No parameters selected

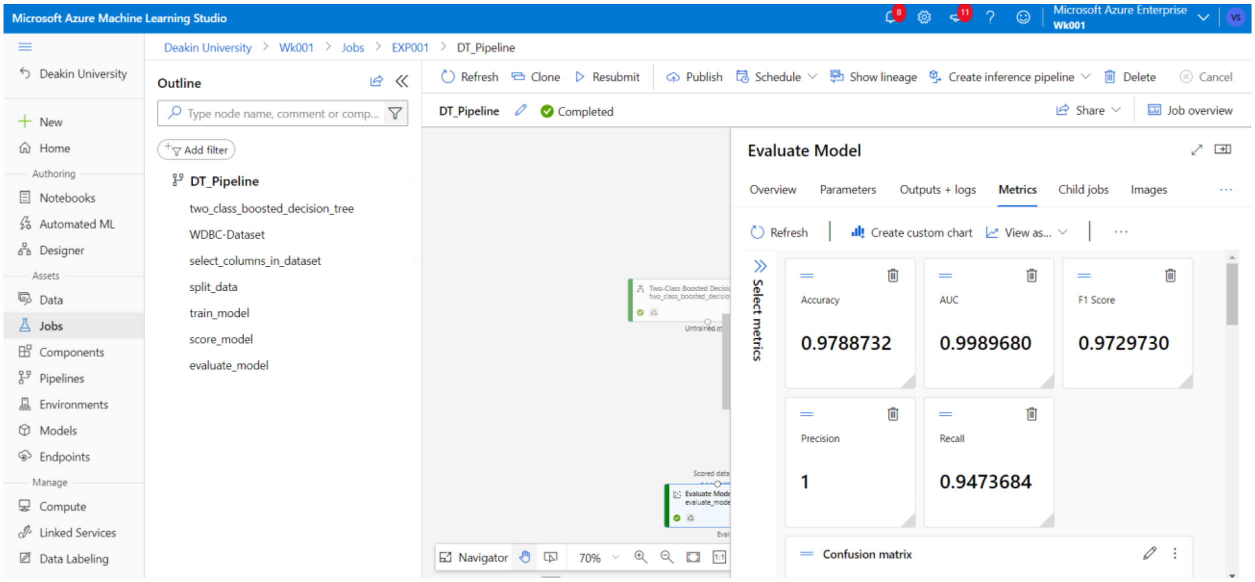
Default datastore

Select datastore *: workspaceblobstore

Before initiating the training process on the dataset, it is essential to choose the ML compute cluster (CC001) that was created in the previous step and utilize it for our training purposes (Khandelwal 2021). Subsequently, we can create a new experiment and click on the "submit" button to commence the model training process.

E. Evaluate the trained model.





Accuracy 0.9788732

AUC 0.9989680

F1 Score 0.9729730

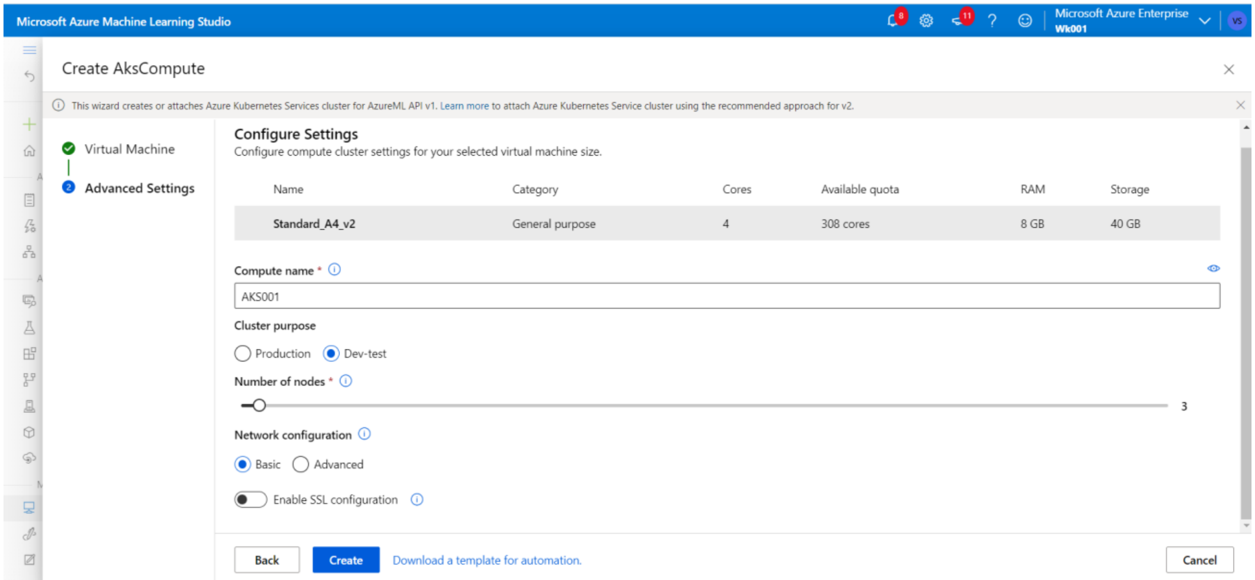
Precision 1

Recall 0.9473684

Confusion matrix

The "Evaluate Model" component in Azure enables users to compare the predicted outputs of a Decision tree model with the actual ground truth labels or target values. This comparison is done using a separate test dataset that was not used for model training. The performance of the model was evaluated using several evaluation metrics, including an accuracy of 0.97, precision of 1, recall of 0.94, F1-score of 0.97, and AUC of 0.99. These metrics provide a quantitative measure of the model's performance in terms of its ability to make accurate predictions, without being biased by the training data.

F. Create Compute for Model Deployment


Microsoft Azure Machine Learning Studio
Microsoft Azure Enterprise
Wk001
Virtual Machine
Advanced Settings
Back
Create
Download a template for automation.
Cancel

Create AksCompute

ⓘ This wizard creates or attaches Azure Kubernetes Services cluster for AzureML API v1. Learn more to attach Azure Kubernetes Service cluster using the recommended approach for v2.

Configure Settings

Configure compute cluster settings for your selected virtual machine size.

Name	Category	Cores	Available quota	RAM	Storage
Standard_A4_v2	General purpose	4	308 cores	8 GB	40 GB

Compute name * ⓘ

AKS001

Cluster purpose

○ Production ● Dev-test

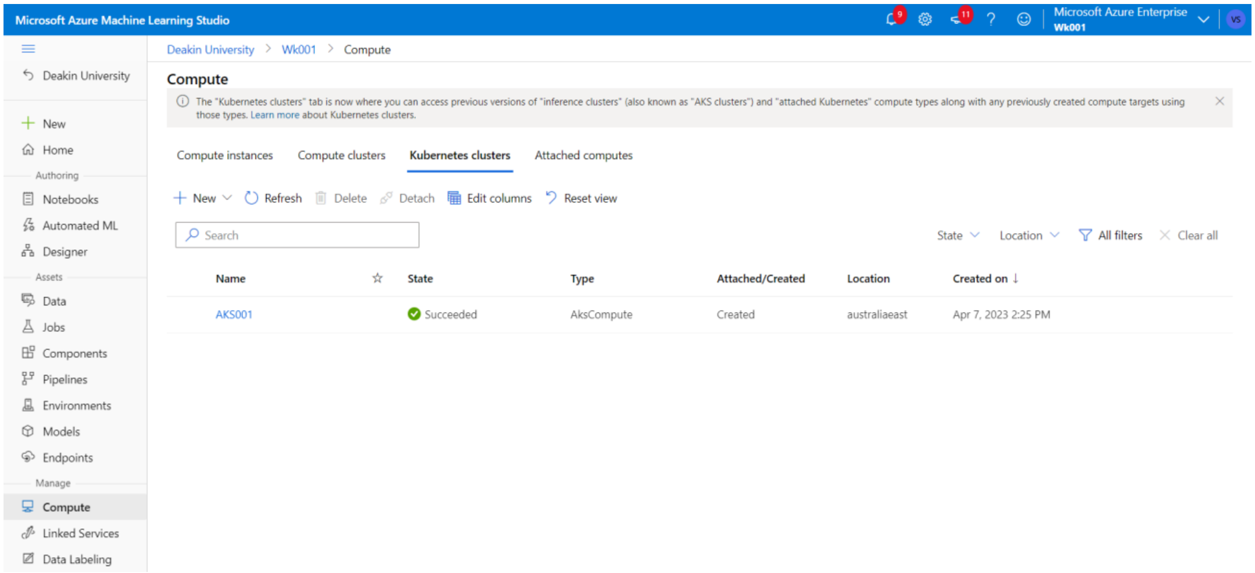
Number of nodes * ⓘ

3

Network configuration ⓘ

● Basic ○ Advanced

Enable SSL configuration ⓘ



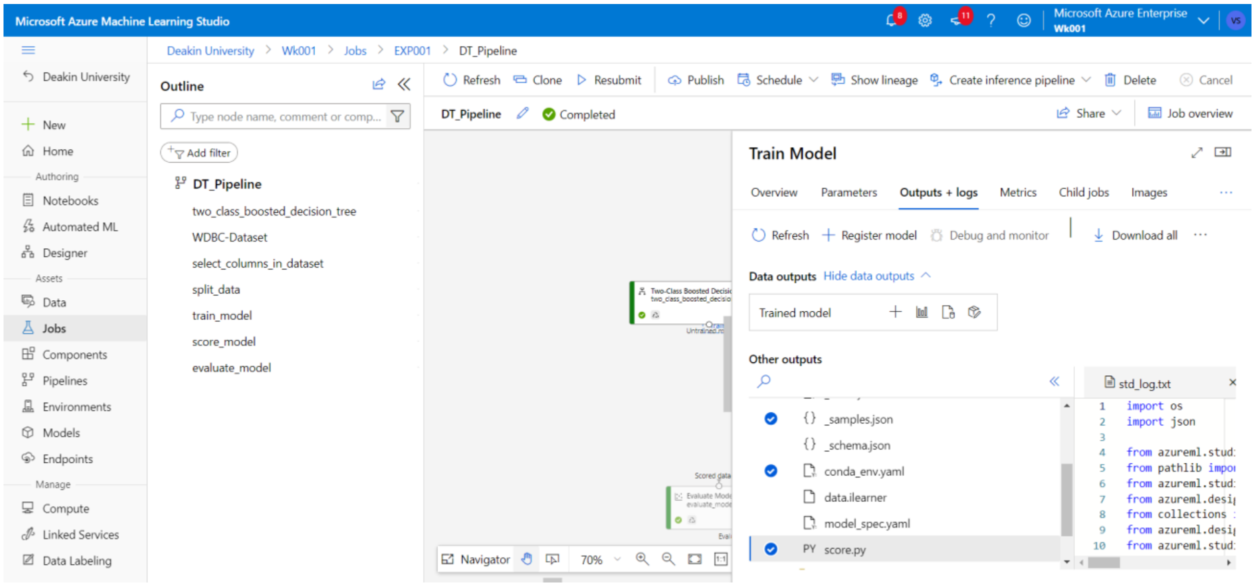
Compute

ⓘ The "Kubernetes clusters" tab is now where you can access previous versions of "inference clusters" (also known as "AKS clusters") and "attached Kubernetes" compute types along with any previously created compute targets using those types. Learn more about Kubernetes clusters.

Name	State	Type	Attached/Created	Location	Created on ↓
AKS001	✓ Succeeded	AksCompute	Created	australiaeast	Apr 7, 2023 2:25 PM

Before implementing the model, it is necessary to establish a computing engine that will ensure the deployment process runs efficiently. In this instance, we have opted to use AKS Compute under Kubernetes Clusters Type. The initial step involves selecting the appropriate virtual machine size for clustering. Next, we assign a name to the Cluster (AKS001) and specify the purpose of the Cluster, which is currently set as a Dev-test. Finally, we set the number of nodes to 3 and create the compute engine for deployment.

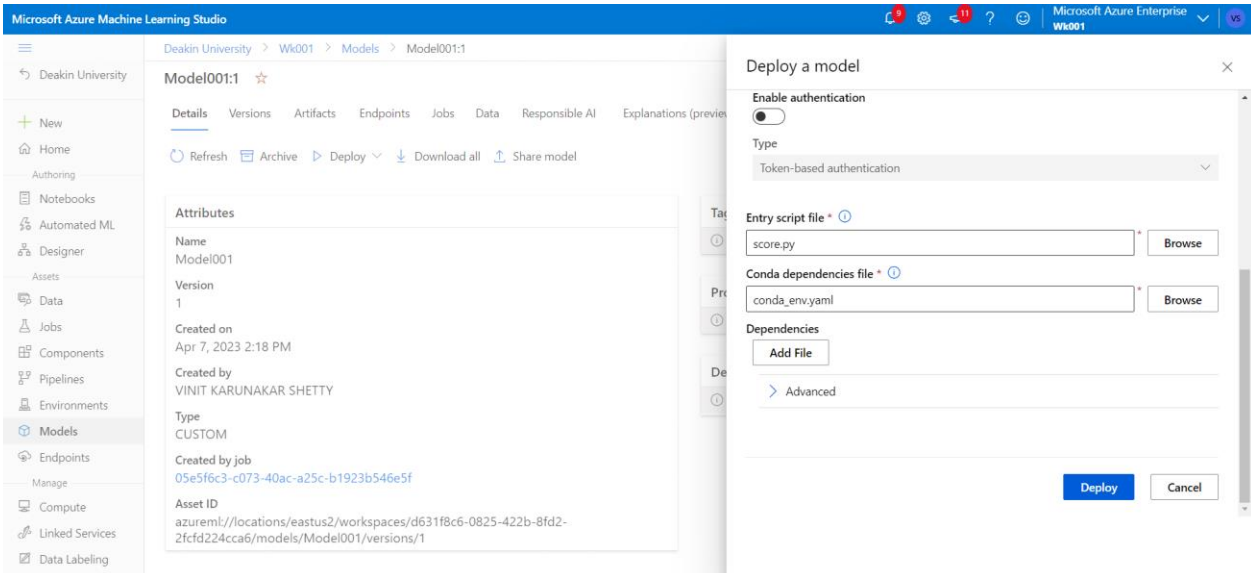
G. Model Deployment



```
import os
import json

from azureml.stud
from pathlib impo
from azureml.stud
from azureml.desi
from collections
from azureml.desi
from azureml.stud
```

To deploy our decision tree model as a Web Service in Azure ML Studio, we needed to upload several files. So, we clicked on the "Train Model" component and accessed the "Outputs + logs" section. From there, we downloaded three files, namely "_Sample.Json", "Conda-env.Yaml", and "Score.py", to our local machine (Rajput 2022).



Attributes

Name
Model001

Version
1

Created on
Apr 7, 2023 2:18 PM

Created by
VINIT KARUNAKAR SHETTY

Type
CUSTOM

Created by job
05e5f6c3-c073-40ac-a25c-b1923b546e5f

Asset ID
azureml://locations/eastus2/workspaces/d631f8c6-0825-422b-8fd2-2fcfd224cca6/models/Model001/versions/1

Enable authentication ○

Type
Token-based authentication

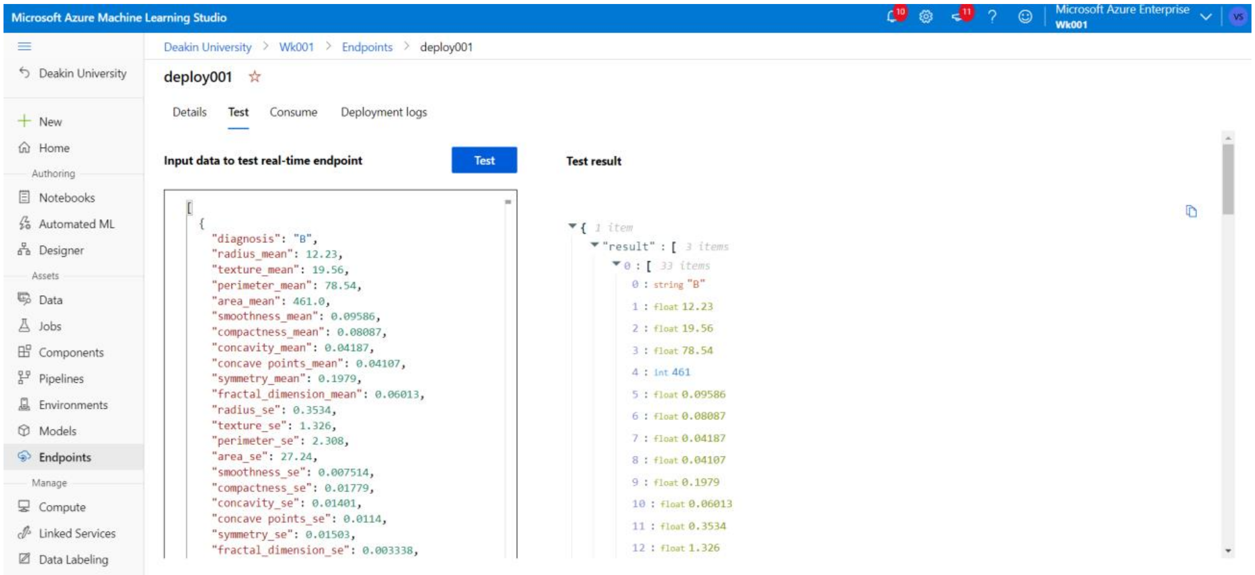
Entry script file *
score.py

Conda dependencies file *
conda_env.yaml

Dependencies

Once the files are downloaded, we choose the "Deploy as Web Service" option from the "Deploy" tab. From there, we select the newly created compute Name (AKS Compute) and Type (AKS001). In the "Entry script file" section, we upload the "Score.py" file, and in the "Conda dependencies" file section, we upload the "Conda-env.Yaml" file that we had previously stored locally. Finally, we click the "deploy" button to initiate the deployment process.

H. Test the deployed model.

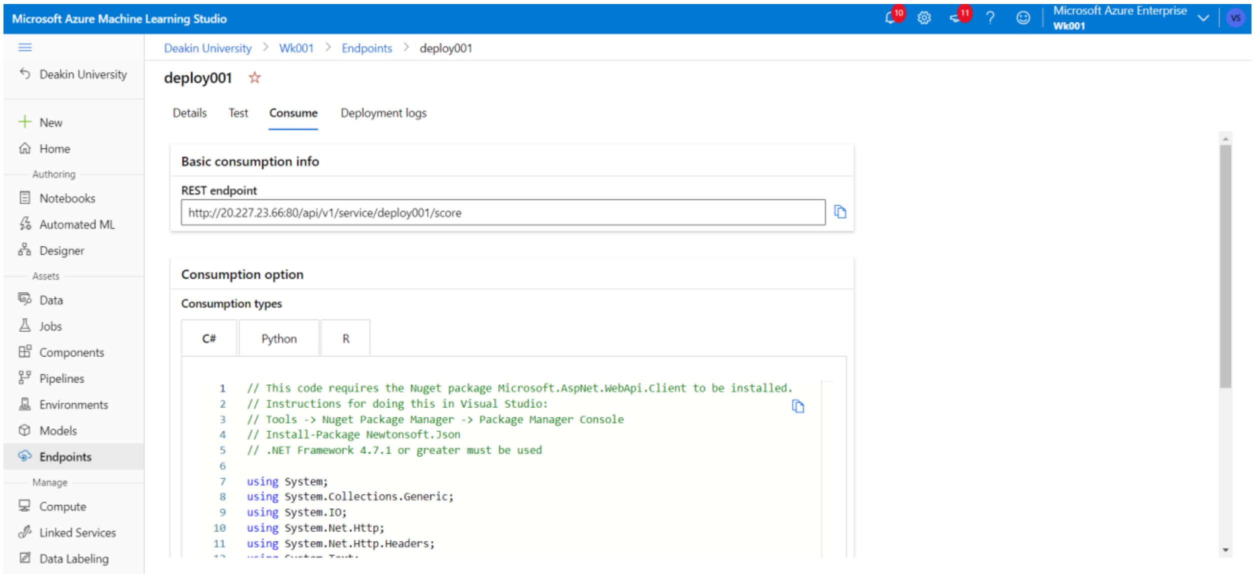


Input data to test real-time endpoint

```
[
  {
    "diagnosis": "B",
    "radius_mean": 12.23,
    "texture_mean": 19.56,
    "perimeter_mean": 78.54,
    "area_mean": 461.0,
    "smoothness_mean": 0.09586,
    "compactness_mean": 0.08087,
    "concavity_mean": 0.04187,
    "concave points_mean": 0.04107,
    "symmetry_mean": 0.1979,
    "fractal_dimension_mean": 0.06013,
    "radius_se": 0.3534,
    "texture_se": 1.326,
    "perimeter_se": 2.308,
    "area_se": 27.24,
    "smoothness_se": 0.007514,
    "compactness_se": 0.01779,
    "concavity_se": 0.01401,
    "concave points_se": 0.0114,
    "symmetry_se": 0.01503,
    "fractal_dimension_se": 0.003338,
```

Test result

```
▾{ 1 item
  ▾"result" : [ 3 items
    ▾0 : [ 33 items
       0 : string "B"
       1 : float 12.23
       2 : float 19.56
       3 : float 78.54
       4 : int 461
       5 : float 0.09586
       6 : float 0.08087
       7 : float 0.04187
       8 : float 0.04107
       9 : float 0.1979
       10 : float 0.06013
       11 : float 0.3534
       12 : float 1.326
```



Basic consumption info

REST endpoint: http://20.227.23.66:80/api/v1/service/deploy001/score

Consumption option

Consumption types

```
// This code requires the Nuget package Microsoft.AspNet.WebApi.Client to be installed.
// Instructions for doing this in Visual Studio:
// Tools -> Nuget Package Manager -> Package Manager Console
// Install-Package Newtonsoft.Json
// .NET Framework 4.7.1 or greater must be used

using System;
using System.Collections.Generic;
using System.IO;
using System.Net.Http;
using System.Net.Http.Headers;
```

Once the deployment status has changed to “Healthy” status, we can proceed to test the real-time endpoint. For that, we use the “_samples.json” file which we had downloaded in the previous step which contains the test data. We upload the JSON file in the text box and click on the “Test” tab and we get the test results of the deployment.

Additionally, when we click on the “Consume” tab we get the REST endpoint URL which can be integrated into the different applications using C#, R and Python (Mishra 2022).

References

1. Frank D (09 March 2022) ‘Improved Machine Learning-Based Predictive Models for Breast Cancer Diagnosis’, *Int. J. Environ. Res. Public Health 2022*, 19(6):3211, doi:10.3390/ijerph19063211
2. George S (25 May 2022) ‘End to End ML pipeline using Azure Machine Learning Studio Designer’, Medium, accessed 07 March 2023.
3. Khandelwal A (28 September 2021) ‘A Comprehensive Guide on Using Azure Machine Learning’, Analytics Vidhya, accessed 08 March 2023.
4. Rajput V (01 October 2022) ‘Deploying pre-trained AI model on Azure‘, Medium, accessed on 08 March 2023.
5. Mishra A (31 October 202) ‘What is Rest API in Python? How to create web APIs with Python?’, Great Learning, accessed on 08 March 2023.