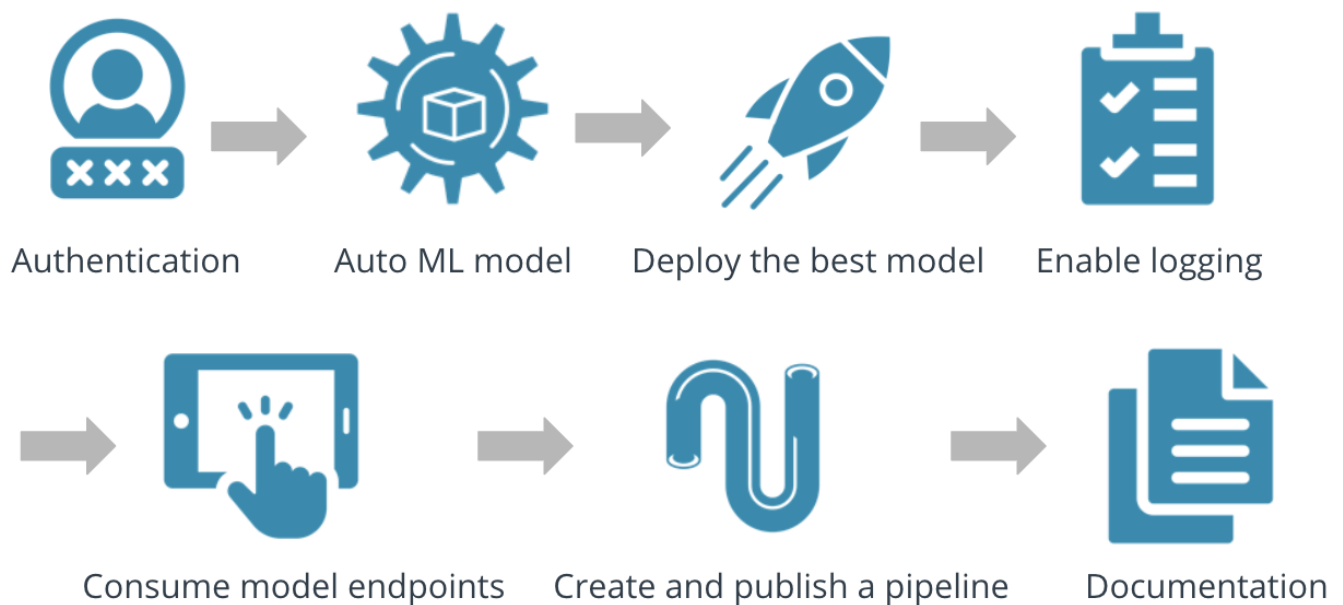


Operationalizing Machine Learning

Overview

This project is part of the Azure ML Nano degree offered by Udacity. Here In this project, we use train a machine learning model with different algorithms using AutoML and also get the best performing model, and operationalize it using below workflow. In this project we demonstrated training of model using first by Azure ML experiment and then by SDK option.

Below figure shows an overview of steps involved:

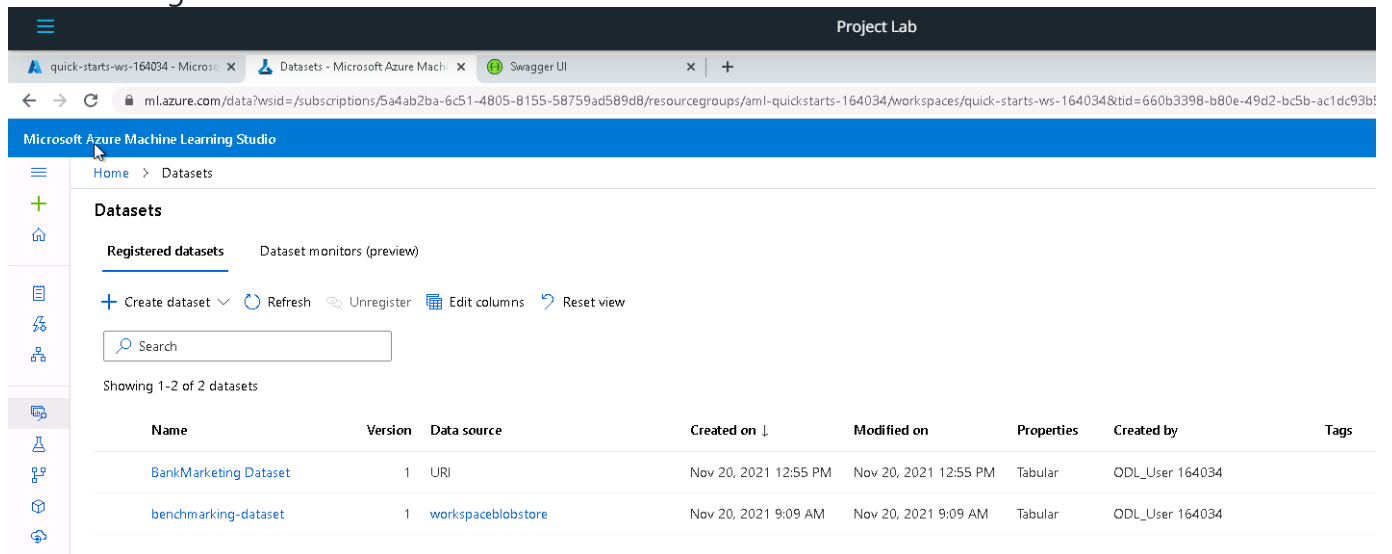


A brief description of each step involved is as follows:

As a first step we need to create a compute instance of running notebook and compute cluster for training the model. Next second step we need to upload dataset and register so that we can start AutoML experiment. Now we are ready to create a new Automated ML experiment as a third step. This experiment gives us the best model best classification model (fourth step). Next we have to deploy the best model (fifth step), once deployment is completed it gives a restful link to interact with the deployed model (sixth step). Now to see the model performance we can enable application insights (seventh step) (also we model testing has to be preform) and model documentation can be viewed using swagger (eighth step) and Model is then consumed (ninth step).

Below are the main steps:

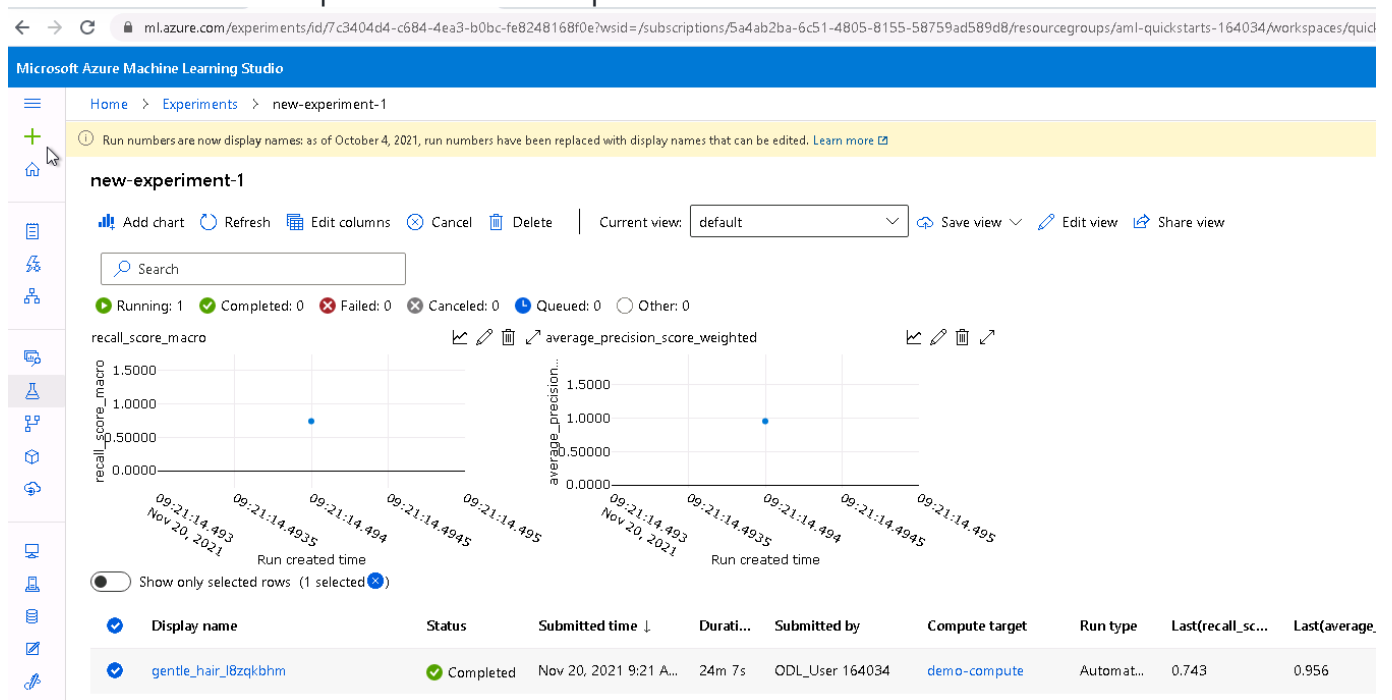
Once we register the dataset it looks like below in azure ml studio:



The screenshot shows the Microsoft Azure Machine Learning Studio interface. The top navigation bar includes 'Home' and 'Datasets'. The left sidebar contains various icons for navigation. The main content area is titled 'Datasets' and shows a list of registered datasets. The 'Registered datasets' tab is active, displaying a table with columns: Name, Version, Data source, Created on, Modified on, Properties, Created by, and Tags. Two datasets are listed: 'BankMarketing Dataset' and 'benchmarking-dataset'.

Name	Version	Data source	Created on	Modified on	Properties	Created by	Tags
BankMarketing Dataset	1	URI	Nov 20, 2021 12:55 PM	Nov 20, 2021 12:55 PM	Tabular	ODL_User 164034	
benchmarking-dataset	1	workspaceblobstore	Nov 20, 2021 9:09 AM	Nov 20, 2021 9:09 AM	Tabular	ODL_User 164034	

The Automated ML experiment after completion will look like this:



The screenshot shows the Microsoft Azure Machine Learning Studio interface for an experiment. The top navigation bar includes 'Home' and 'Experiments'. The left sidebar contains various icons for navigation. The main content area is titled 'new-experiment-1' and shows a summary of the experiment's status and results. The 'Run numbers are now display names' message is visible. The 'new-experiment-1' section displays a table with columns: Display name, Status, Submitted time, Duration, Submitted by, Compute target, Run type, Last(recall_score_macro), and Last(average_precision_score_weighted). One experiment is listed: 'gentle_hair_l8zqkbhm' with a status of 'Completed'.

Display name	Status	Submitted time	Duration	Submitted by	Compute target	Run type	Last(recall_score_macro)	Last(average_precision_score_weighted)
gentle_hair_l8zqkbhm	Completed	Nov 20, 2021 9:21 A...	24m 7s	ODL_User 164034	demo-compute	Automat...	0.743	0.956

The best performing model is the Voting Ensemble:

ml.azure.com/automl/experiments/id/7c3404d4-c684-4ea3-b0bc-fe8248168f0e/rundetails/AutoML_620fd52d-348f-460b-abcb-aa9ba81cbce8?wsid=/subscriptions/5a4ab2ba-6c51-4805-8155-58759ad589d8/resourcegroups/aml-q

Microsoft Azure Machine Learning Studio

Home > Automated ML > new-experiment-1 > gentle_hair_l8zqkbhm

gentle_hair_l8zqkbhm

Refresh Cancel Delete

Details Data guardrails **Models** Outputs + logs Child runs Snapshot

Refresh Deploy Download Explain model Edit columns Reset view

Search

Showing 1-25 of 69 models

Algorithm name	Explained	AUC weighted ↓	Sampling	Submitted time	Duration	Hyperparameter
VotingEnsemble	View explanation	0.94758	100.00 %	Nov 20, 2021 9:43 AM	1m 41s	algorithm : [XGBoostClassifier, ...
StackEnsemble		0.94678	100.00 %	Nov 20, 2021 9:43 AM	2m 10s	algorithm : [XGBoostClassifier, ...
SparseNormalizer, XGBoostClassifier		0.94587	100.00 %	Nov 20, 2021 9:34 AM	32s	booster : gbtrees colsample_bytr ...
StandardScalerWrapper, XGBoostClassifier		0.94529	100.00 %	Nov 20, 2021 9:40 AM	2m 2s	booster : gbtrees colsample_bytr ...
StandardScalerWrapper, XGBoostClassifier		0.94521	100.00 %	Nov 20, 2021 9:40 AM	1m 46s	booster : gbtrees colsample_bytr ...
MaxAbsScaler, LightGBM		0.94505	100.00 %	Nov 20, 2021 9:23 AM	12s	min_data_in_leaf : 20 ...

Page 1 of 3

After enabling application insight the flag turn true as shown below:

Project Lab

quick-starts-ws-164034 - Microsc best-model-for-deploy - Microsc kalimi03/nd00333_AZMLND_C2

ml.azure.com/endpoints/lists/realtimeendpoints/best-model-for-deploy/detail?wsid=/subscriptions/5a4ab2ba-6c51-4805-8155-58759ad589d8/res

Microsoft Azure Machine Learning Studio

Home > Endpoints > best-model-for-deploy

best-model-for-deploy

Details Test Consume Deployment logs

Created on
11/20/2021 9:53:54 AM

Last updated on
11/20/2021 10:25:33 AM

Image ID
--

REST endpoint
<http://4e9e3a85-ddc1-491d-b13e-001315e2c390.southcentralus.azurecontainer.io/score>

Key-based authentication enabled
true

Swagger URI
<http://4e9e3a85-ddc1-491d-b13e-001315e2c390.southcentralus.azurecontainer.io/swagger.json>

CPU
1.8

Memory
4 GB

Application Insights enabled
true

Application Insights url
<https://portal.azure.com#resource/subscriptions/5a4ab2ba-6c51-4805-8155-58759ad589d8/resourcegroups/aml-q/quickstarts-164034/providers/microsoft.insights/components/mlappinsight164034>

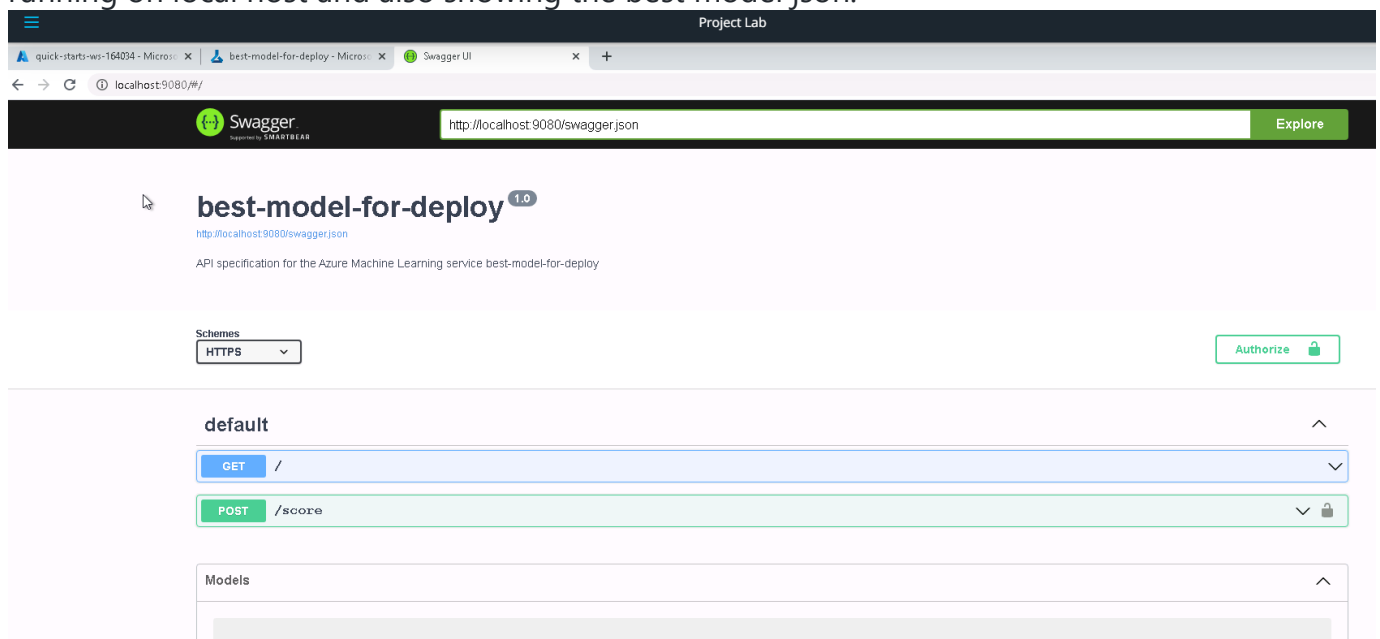
Below is the screen shot for result from running log.py file content from notebook:

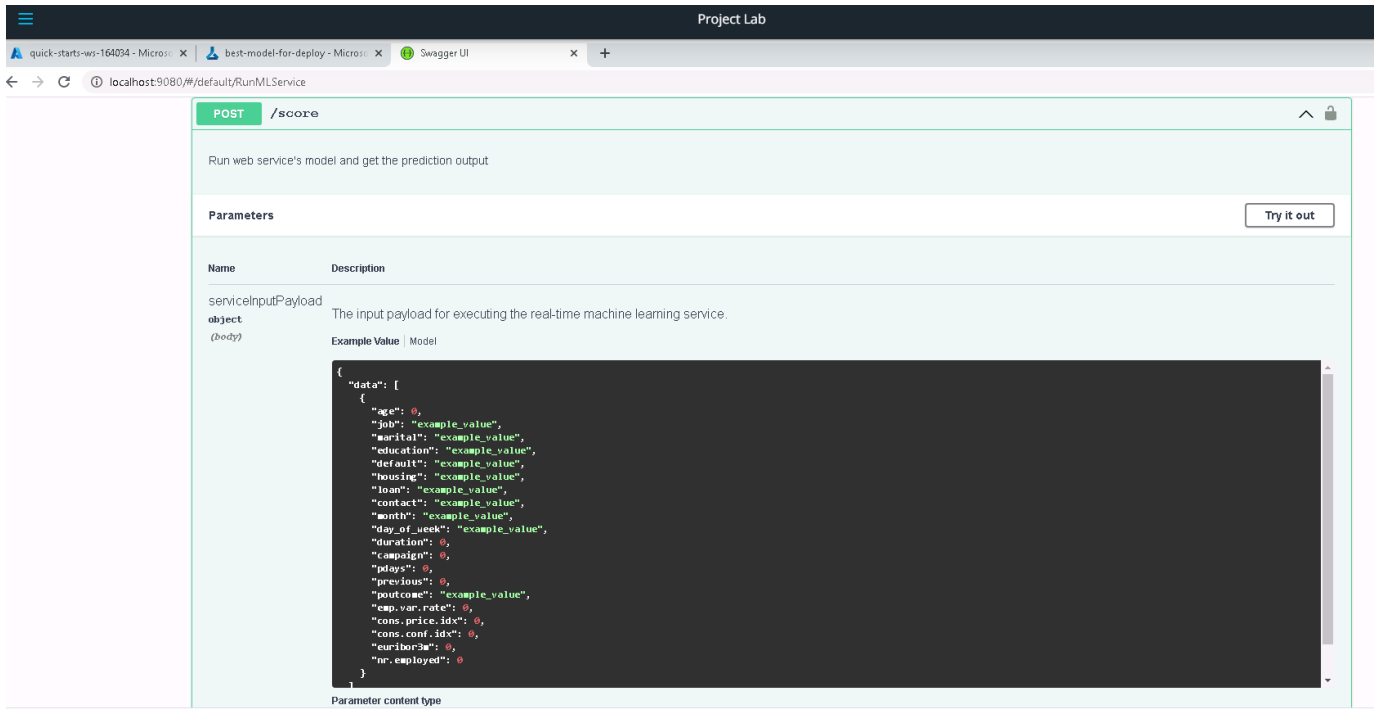
```
1 from azureml.core import Workspace
2 from azureml.core.webservice import Webservice
3
4 ws = Workspace.from_config()
5 service = Webservice(name = "best-model-for-deploy", workspace=ws )
6
7 logs = service.get_logs()
8 for line in logs.split('\n'):
9     print(line)
10
11 service.update(enable_app_insights = True)
```

[7] ✓ 3 sec

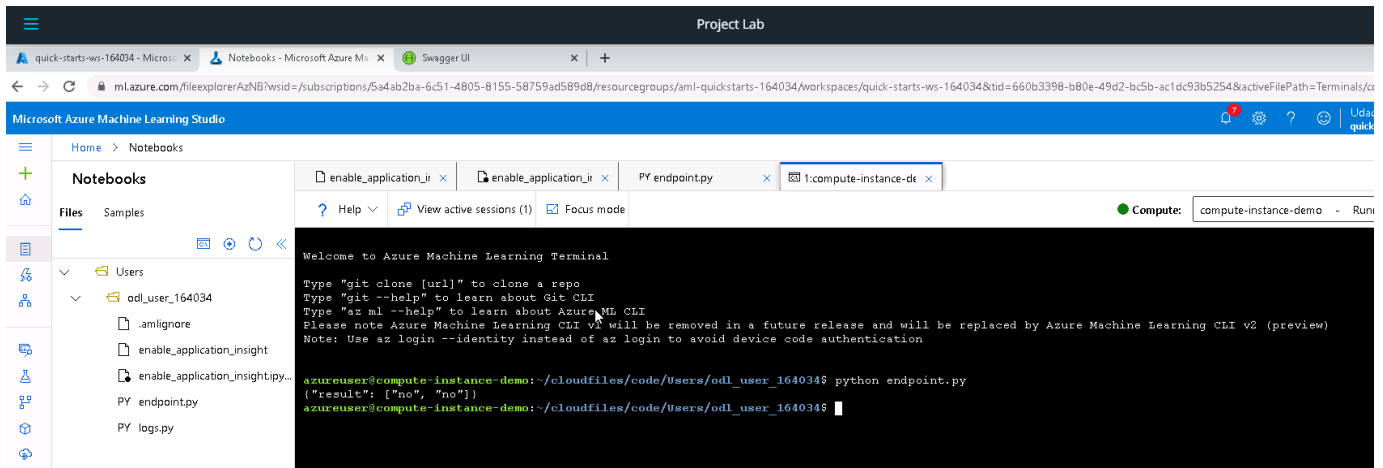
... SPARK_HOME not set. Skipping PySpark Initialization.
Generating new fontManager, this may take some time...
Initializing logger
2021-11-20 10:08:13,707 | root | INFO | Starting up app insights client
logging socket was found. logging is available.
logging socket was found. logging is available.
2021-11-20 10:08:13,708 | root | INFO | Starting up request id generator
2021-11-20 10:08:13,708 | root | INFO | Starting up app insight hooks
2021-11-20 10:08:13,708 | root | INFO | Invoking user's init function
2021-11-20 10:08:17,340 | root | INFO | Users's init has completed successfully
2021-11-20 10:08:17,345 | root | INFO | Skipping middleware: dbg_model_info as it's not enabled.
2021-11-20 10:08:17,345 | root | INFO | Skipping middleware: dbg_resource_usage as it's not enabled.
Generating swagger file: /tmp/tmpaquhsmb1
2021-11-20 10:08:17,350 | root | INFO | Scoring timeout is found from os.environ: 60000 ms
2021-11-20 10:08:22,698 | root | INFO | 200
127.0.0.1 - - [20/Nov/2021:10:08:22 +0000] "GET /swagger.json HTTP/1.0" 200 3564 "-" "Go-http-client/1.1"
2021-11-20 10:10:27,437 | root | INFO | 200

Swagger is basically used for communicating rest apli endpoint documentation a json file. The bash file swagger.sh is used to run a swagger docker image and serve.py make swagger json file available on an local host server. Below figures shows docker image (or swagger-ui.html) running on local host and also showing the best model json:





We use `endpoint.py` file for deployed model consumption. When a json format document is passed on to the rest api endpoint using `POST` method, then the model processes it and gives the outcome as show below:



Now creating a pipeline for the entire process using Jupyter Notebook:
 “aml-pipelines-with-automated-machine-learning-step.ipynb”

Microsoft Azure Machine Learning Studio

Home > Pipelines

Pipelines

Pipeline runs Pipeline endpoints Pipeline drafts

Run numbers are now display names: as of October 4, 2021, run numbers have been replaced with display names that can be edited. [Learn more](#)

+ New pipeline Refresh Edit columns Reset view

Search

Status Experiment

Showing 1-2 of 2 runs

Display name	Experiment	Status	Description	Submitted time ↓	Duration	Submitted by	Tags
silver_soccer_bgcr229	pipeline-rest-endpoint	Completed		Nov 20, 2021 1:08 PM	4s	ODL_User 164034	
bubbly_loquat_bp1jf0km	new-experiment-1	Completed	pipeline_with_automist...	Nov 20, 2021 12:56 PM	9m 50s	ODL_User 164034	

An endpoint is created using pipeline as shown below:

Microsoft Azure Machine Learning Studio

Home > Experiments > pipeline-rest-endpoint > silver_soccer_bgcr229

silver_soccer_bgcr229

Refresh Clone Publish Resubmit Cancel Delete

Graph Steps Outputs + logs Metrics Images Snapshot Explanations (preview) Fairness (preview)

Search canvas

BankMarketing Dataset

Dataset output

training_data

automl_module

Completed

metri... model... defau... defau...

Navigator 100%

Pipeline run overview

Attributes Pipeline parameters

Run ID
d01e266e-01f8-467f-84b8-34946afc2242

Run source
Unavailable

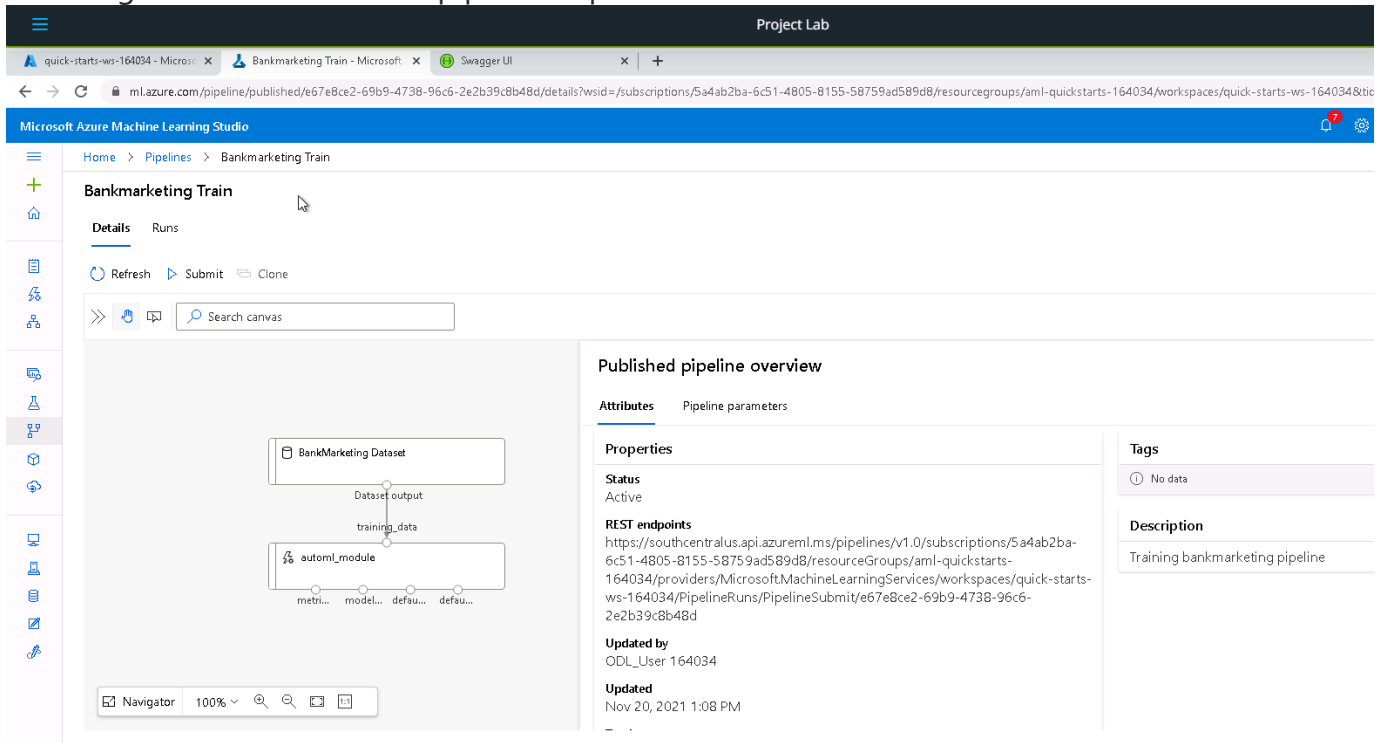
Run type
HTTP

Published pipeline
[View detail](#)

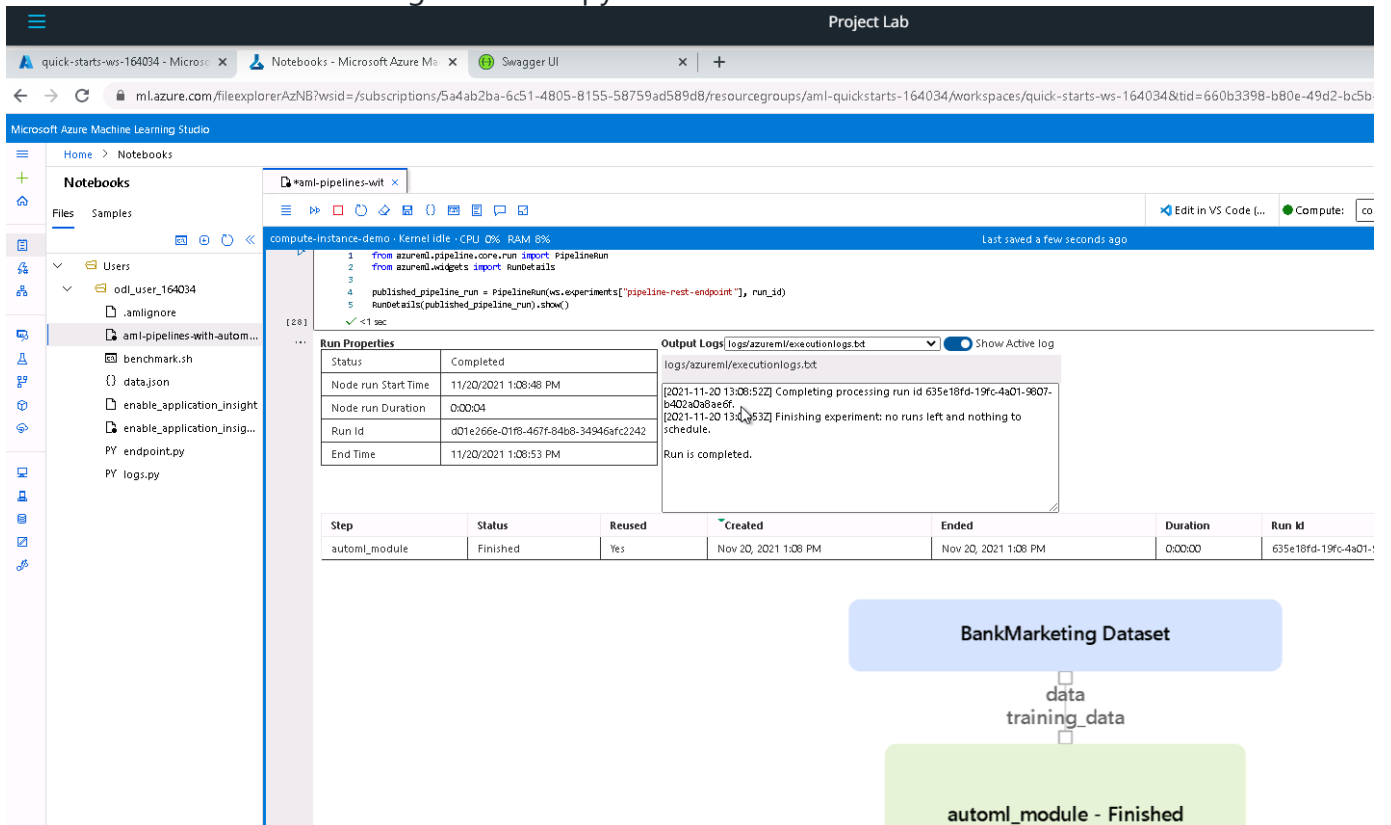
Experiment
[pipeline-rest-endpoint](#)

Submit time
Nov 20, 2021 1:08 PM

Below figure shows state after pipeline is published:



Outcome of Run Details Widget in the Jupyter Notebook is show below:



Further work:

1. Use of Deep Learning: This may provide a better performance, since image pixel correlation information is better utilized in case deep learning.
2. Increasing the dataset size will also help.
3. Adding a CI/CD pipeline to interact with the Published Pipeline will be useful
4. Maybe data preprocessing (in case of deep learning) will boost the performance.

Screen Recording

You can view the screencast of all the steps mentioned above at this:

<https://www.youtube.com/watch?v=6afnf27xvfl>