



Tutorial for Machine Learning Pipeline

OEA Curated

Date Published: December 2021

This document's content is governed by [Microsoft's Legal Notices](#) for open source contributions.



Table of Content

I.	Run a sample pipeline	4
1.	Preliminaries	4
2.	Setup Azure environment	4
2.1	Create Azure Machine Learning (AML) workspace	4
2.2	Configure access control in Azure Machine Learning	5
2.3	Create a linked service in Azure Synapse Analytics	8
3.	Setup Pipeline Template	10
3.1	Upload pipeline template	10
3.2	Update configuration	11
3.3	Run pipeline	11
4.	Check Dataset in Synapse	12
4.1	Data (stage1p)	12
4.2	Data (stage2p & 2np)	12
4.3	Data (stage2p/Processed)	12
4.4	Data (stage3p)	13
5.	Check training result in AML	14
5.1	Data (AML)	14
5.2	Experiment	14
5.3	Source Code	16
5.4	Model	16
6.	Run error analysis	16
II.	Run your own experiment	18
1.	Explore data	18
2.	Engineer features	18
2.1	Create featurization notebook	18
2.2	Include notebook into pipeline	18
3.	Re-create main dataset	18
4.	Change dataset for training	18
5.	Submit an experiment to AML	19
6.	Run AutoML	19
7.	Add a new model to AML	20



7.1	Setup development environment.....	20
7.2	Submit experiment with default configuration	20
7.3	Implement model.....	21
7.4	Train the new model from Azure Synapse.....	23



I. Run a sample pipeline

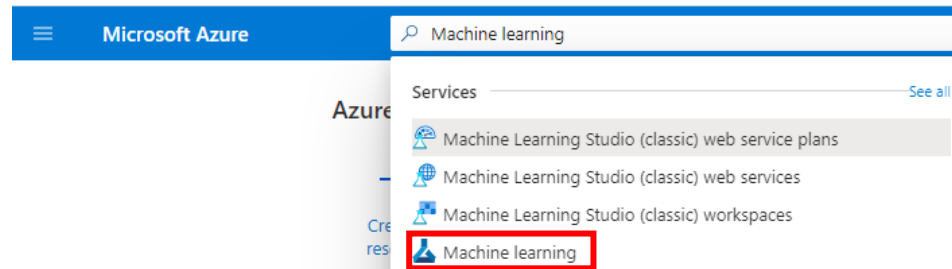
1. Preliminaries

- Environment with OpenEduAnalytics
 - To setup an environment, follow the steps in [OpenEduAnalytics/README.md at main · microsoft/OpenEduAnalytics \(github.com\)](https://github.com/microsoft/OpenEduAnalytics/blob/main/README.md)

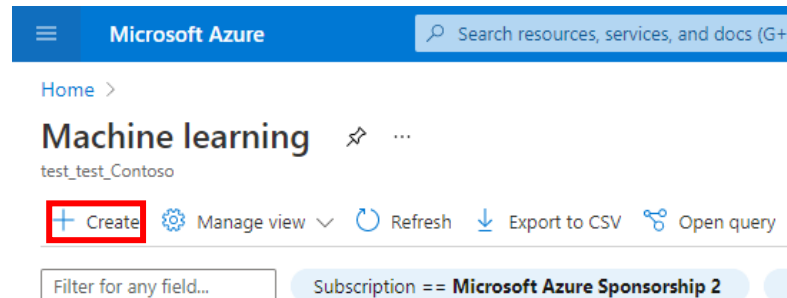
2. Setup Azure environment

2.1 Create Azure Machine Learning (AML) workspace

- (1) On Azure Portal, select “Machine learning”



- (2) Click “+ Create”



- (3) Enter “Workspace name” and “Region” as you prefer, chose “Key vault” and “Application insights” under your resource group, create a new container registry, then click “Review + create”



Microsoft Azure

Home > syn-aea-cisd3mlmod1 > rg-aea-cisd3mlmod1 > Create a resource > Machine Learning >

Machine learning

Create a machine learning workspace

Basics Networking Advanced Tags Review + create

Project details
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource group *

Workspace details
Specify the name and region for the workspace.

Workspace name *

Region *

Storage account *

Key vault *

Application insights *

Container registry *

Create new container registry

Name *

SKU *

Review + create Previous Next: Networking Save Discard

- (4) Open Azure Machine Learning Studio. Click “Compute” tab > “+New” and create a compute with the name you like (This information will be necessary later)

Microsoft Azure Machine Learning Studio

Home > Compute

Compute

Compute instances Compute clusters Inference clusters

+ New Refresh Start Stop Restart

Search

Name	State
mlcompute1117	Running

2.2 Configure access control in Azure Machine Learning

- (1) On Azure portal, open AML workspace created in previous step and click “Access control (IAM)” > “+ Add” > “Add role assignment”



(2) Choose “Contributor” and click “Next”

(3) On “Members” tab,

- Select “Managed identity”
- Click “+ Select members”

On “Select managed identities” pane

- Select “Synapse workspace” as “Managed Identity”
- Enter the name of your Synapse workspace and search for it
(An icon of your Synapse workspace will be displayed)
- Click the icon of your Synapse workspace
- Click “Select” button



Microsoft Azure Search resources, services, and docs (G+)

Home > ml_mlmod1 >

Add role assignment

Got feedback?

Role **Members** Review + assign

Selected role
Contributor

Assign access to
☐ User, group, or service principal
☒ **Managed identity**

Members
+ Select members

Name	Object ID	Type
No members selected		

Description
Optional

Review + assign Previous Next

Select managed identities

Got feedback?

Subscription *
Microsoft Azure Sponsorship 2

Managed identity
Synapse workspace (43)

Select
syn-oea-cisd3mlmod1

syn-oea-cisd3mlmod1
/subscriptions/7b9a4896-4541-483f-bdc7-d8f4ec6be3ee/resourceGroups/rg-oea-...
Showing 1 of 43 results. Try to narrow your search.

Selected members:
No members selected. Search for and add one or more members you want to assign to the role for this resource.
[Learn more about RBAC](#)

Select Close

(4) Click "Review + assign" on "Members" tab and then on "Review + assign" tab



Add role assignment ...

[Got feedback?](#)

Role Members Review + assign

Selected role

Contributor

Assign access to

- ☐ User, group, or service principal
☒ Managed identity

Members

[+ Select members](#)

Name	Object ID	Type
syn-oea-cisd3mlmod1	cbac0d54-1d58-4310-9745-0e19c13e26...	Synapse workspace ⓘ

Description

Optional

[Review + assign](#) [Previous](#) [Next](#)

Add role assignment ...

[Got feedback?](#)

Role Members Review + assign

Role

Contributor

Scope

/subscriptions/7b9a4896-4541-483f-bdc7-d8f4ec6be3ee/resourceGroups/rg-oea-cisd3mlmod1/providers/Microsoft.MachineLearningServices/

Members

Name	Object ID	Type
syn-oea-cisd3mlmod1	cbac0d54-1d58-4310-9745-0e19c13e26ed	Synapse workspace ⓘ

Description

No description

[Review + assign](#) [Previous](#)

2.3 Create a linked service in Azure Synapse Analytics

(1) On Synapse Studio, click "Manage" tab > "Linked Services" > "+ New"

Microsoft Azure | Synapse Analytics | syn-oea-cisd3mlmod1

>> / yuki branch Validate all Commit all Publish

Analytics pools

- SQL pools
- Apache Spark pools
- Data Explorer pools (preview)

External connections

- Linked services**
- Azure Purview

Integration

Linked services

Linked services are much like connection strings, which define the external resources. [Learn more](#)

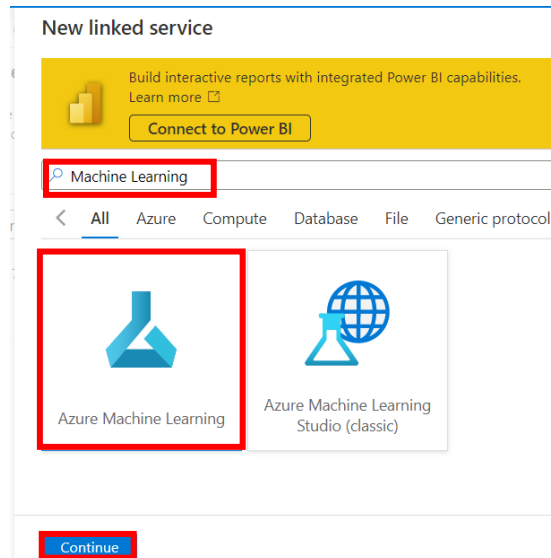
[+ New](#)

Filter by name Annotations : Any

Showing 1 - 7 of 7 items

Name	Type
LS_ADLS_OEA	Azure Data Lake Storage Gen2

(2) Enter "Machine Learning" as a search key, select the icon, and click "Continue"



- (3) Do the following and click “Commit”
- Enter “Name” as you prefer
 - Select “Managed Identity” as “Authentication method”
 - Select your “Azure subscription”
 - Select your “Azure Machine Learning workspace name” as you have created in 2.1(3)
 - Click “Test connection” and make sure you see “Connection successful”
 - Click “Commit” button



New linked service (Azure Machine Learning)

Choose a name for your linked service. This name cannot be updated later.

Name *
LS_ML_OEA

Description

Connect via integration runtime *
AutoResolveIntegrationRuntime

Authentication method
Managed Identity

Azure Machine Learning workspace selection method
☒ From Azure subscription ☐ Enter manually

Azure subscription
Microsoft Azure Sponsorship 2 (7b9a4896-4541-483f-bdc7-d8f4ec6be3ee)

Azure Machine Learning workspace name *
ml_mlmod1

Managed identity name: **syn-oea-cisd3mlmod1**
Managed identity object ID: **cbac0d54-1d58-4310-9745-0e19c13e26ed**
Grant workspace service managed identity access to your Azure Machine Learning.
[Learn more](#)

Annotations
+ New

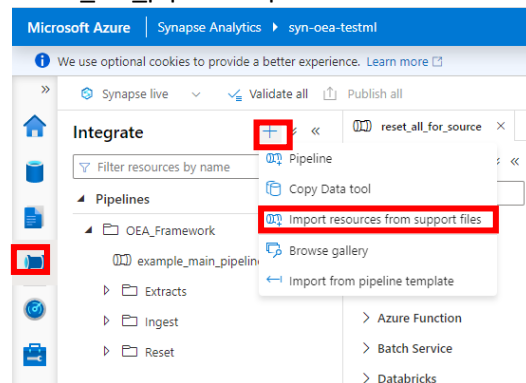
> Advanced

Commit Back Connection successful
Test connection Cancel

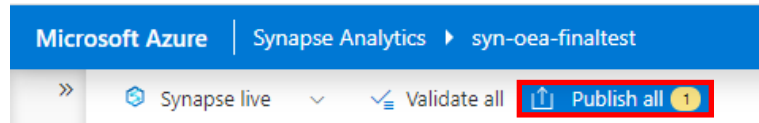
3. Setup Pipeline Template

3.1 Upload pipeline template

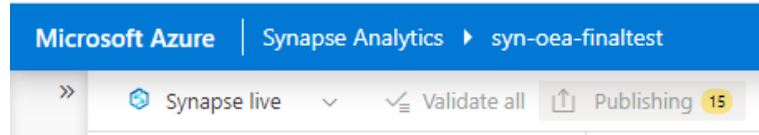
- (1) Click “Integrate” tab > “+” > “Import resources from support files” and upload “Data_ML_pipeline.zip”



- (2) Click “Publish all” and



(3) Make sure



3.2 Update configuration

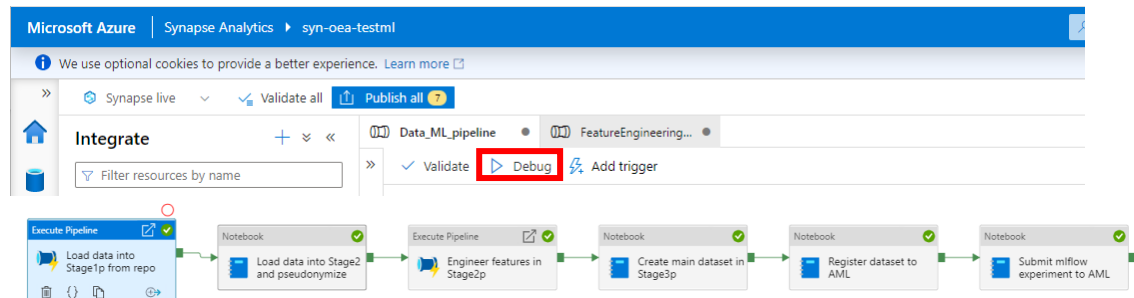
- (1) Open “ml_config” notebook and modify the following values
 - LINKED_SERVICE_NAME Specify the name of the linked service you created in 2.3
 - COMPUTE Specify the name of the compute created in 2.1(4)

Configuration for AML

```
1 ML_CONFIG = {
2     "key": {"PersonId_pseudonym"},           # Key columns in dataset
3     "feature": None,                         # Feature columns in dataset (When None, all the features except label and exclude_feature will used)
4     "exclude_feature": None,                 # Columns in dataset to be excluded from features
5     "sensitive_feature": {"GenerationCode"}, # Sensitive feature columns in dataset (When None or {}, no dataset for sensitive features will be re
6     "label": {"avg_numeric_grade"},          # Label columns in dataset
7     "label_dtype": "double",                 # Data type of label
8     "model": "regression_linearreg",          # Model class name
9 }
10 MODEL_VERSION = '1'                        # Version of trained model used for prediction (Version can be checked in "Models" tab in AML)
11                                             # "predict" feature requires to specify version (it doesn't have feature to use the latest version)
12 LINKED_SERVICE_NAME = "LS_ML"               # Link Service from Synapse to AML
13 COMPUTE = "testcluster"                     # Compute in AML
14
15 URI = "https://github.com/lladap/OpenEduAnalytics/packages/How_to_implement_Azure_machine_learning/aml_modeling/project" # URI to the mlflow project in Github repo
16 BRANCH = "maidap-fy22h1-mlpipeline"        # Branch of Github repo
17
18 # When repo is private, URI needs to be set with personal access token as shown below
19 #URI = f"https://{TOKEN}@github.com:ContosoISD3/cisdggimpl5.git#aml_projects/mlflow/template_project/project/"
20 # Personal Access Token (No need to modify this for now.
21 # https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token)
```

3.3 Run pipeline

- (1) Click “Commit all” and “Publish”
- (2) Click “Debug” and run the pipeline and make sure all activities are completed without errors as shown below



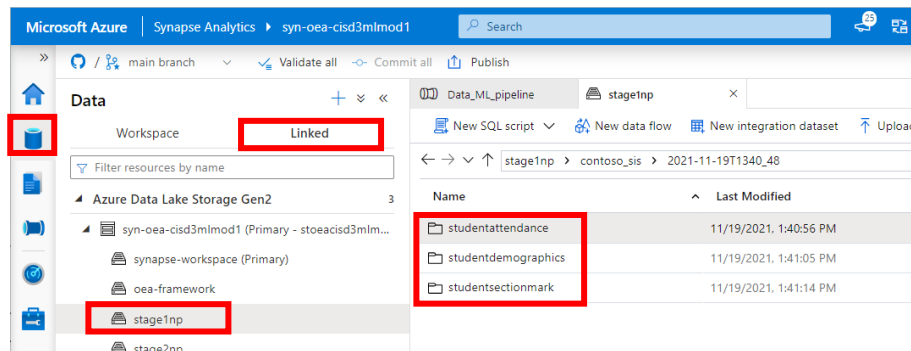


4. Check Dataset in Synapse

4.1 Data (stage1p)

First activity in the pipeline (“Load data into Stage1p - copy test data from URL”) imports data from Github repo to stage1np.

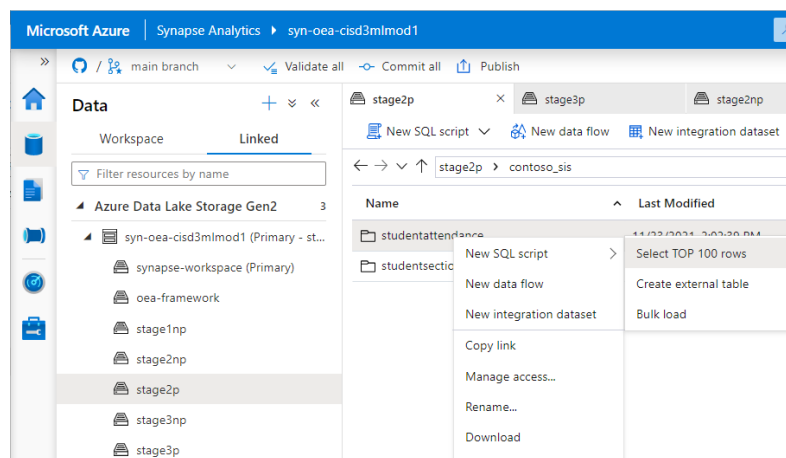
In Synapse, click “Data” tab and check the imported data in stage1np/contoso_sis/[datetime stamp] and stage1np/m365/[datetime stamp] as shown below. Csv files are downloaded from [Github repo](#) and saved in those folders.



4.2 Data (stage2p & 2np)

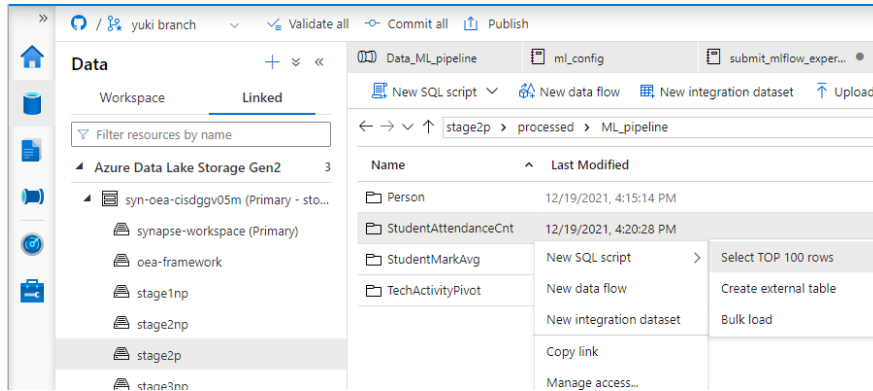
Second activity in the pipeline (“Load data into Stage2 and pseudonymize”) imports the data in stage1np to stage2p and 2np.

Check “Delta” data imported from stage in stage2p/contoso_sis/(also stage2np/contoso_sis/) and stage2p/m365/ as shown below.



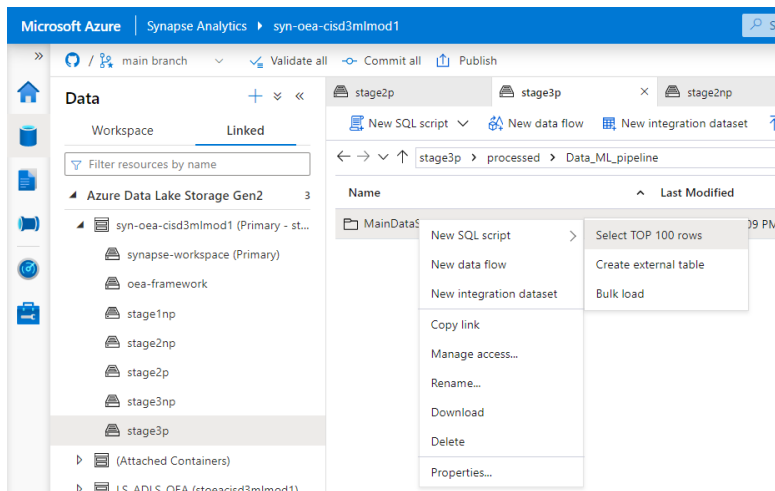
4.3 Data (stage2p/Processed)

3rd activity in the pipeline (“Engineer features”) creates features for each student and save it as “Delta” data in stage2p/processed/ML_pipeline/.

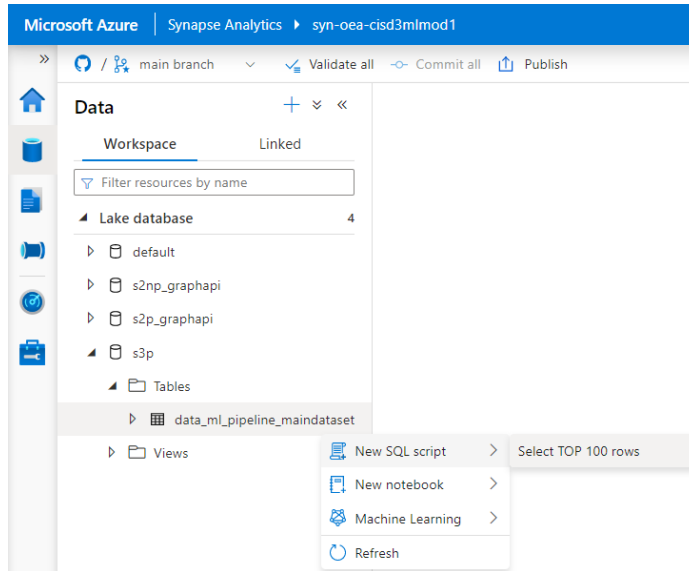


4.4 Data (stage3p)

4th activity in the pipeline (“Create main dataset in Stage3p”) joins the datasets created in the previous step and saves it as “Delta” in stage3p/ML_pipeline/MainData. This dataset has all the possible features, key, and labels for machine learning.



A corresponding datatable is also created in Lake database named “s3p”.

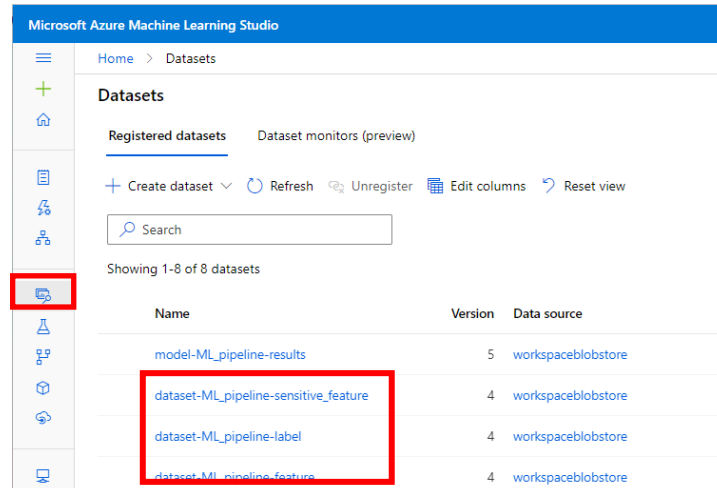


5. Check training result in AML

5.1 Data (AML)

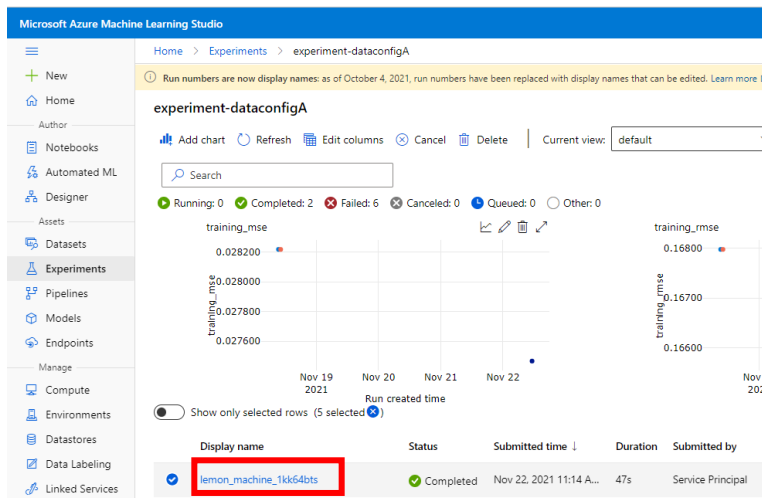
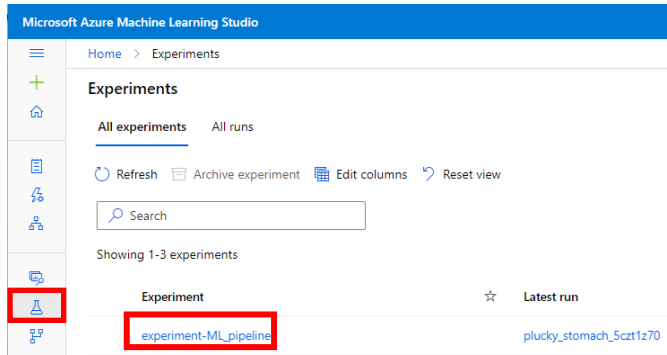
- (1) 5th activity in the pipeline (“Register dataset to AML”) registers 3 datasets to AML. These are the subsets of main dataset created previously.

In AML studio, click “Datasets” tab. Then you see 3 datasets registered as shown as below. These are “features”, “label”, and “sensitive_feature”.

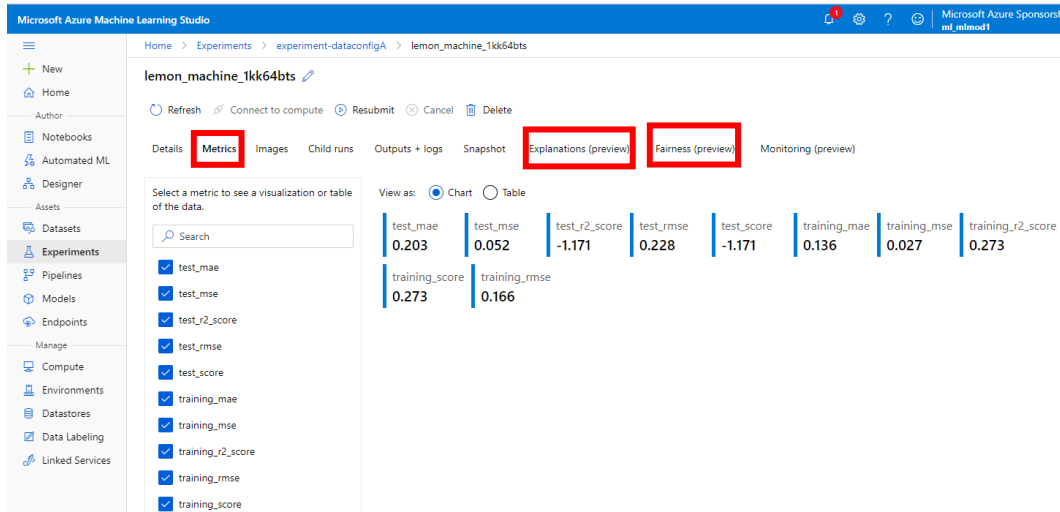


5.2 Experiment

- (1) Click “Experiments” tab > “experiment-ML_pipeline”> “[random display name as shown below]”. This is the experiment the last activity in Synapse pipeline has submitted.



(2) Check evaluation results by clicking “Metrics”, “Explanations”, and “Fairness”



You can find details in the webpages below

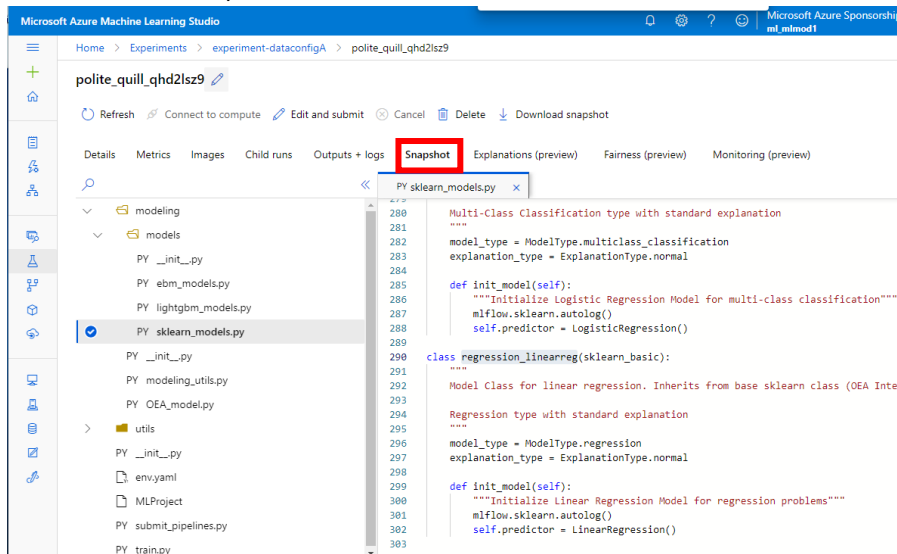
- Explanations [Use Python to interpret & explain models \(preview\) - Azure Machine Learning | Microsoft Docs](#)



- Fairness [Assess ML models' fairness in Python \(preview\) - Azure Machine Learning | Microsoft Docs](#)

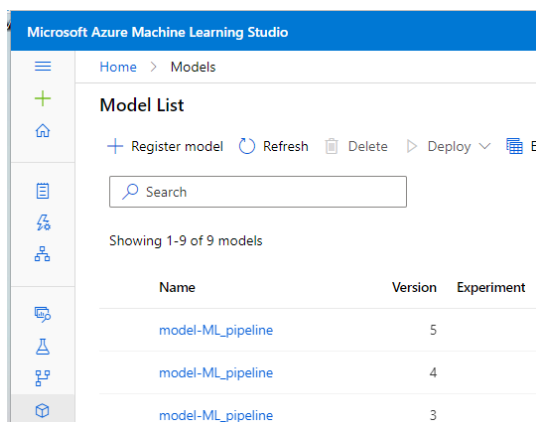
5.3 Source Code

- (1) Click “Experiments” tab > “experiment-dataconfigA” > “[random display name as shown below]” > “Snapshot” to see the source code downloaded from Github repository. (You can also find the source code by opening the link to the repository in “ml_config” notebook.)



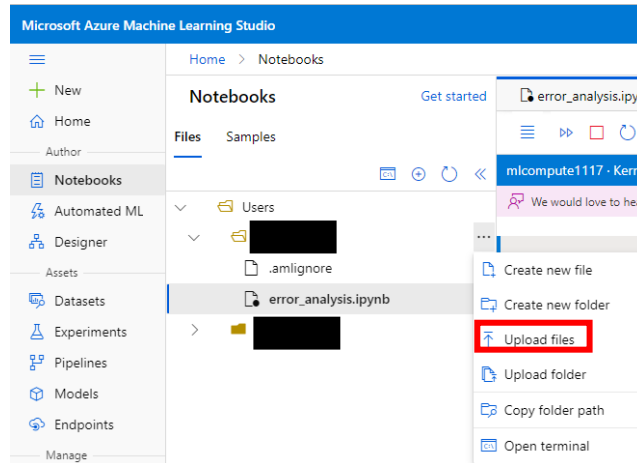
5.4 Model

- (1) Click “Models” > “model-ML-pipeline” for the trained model

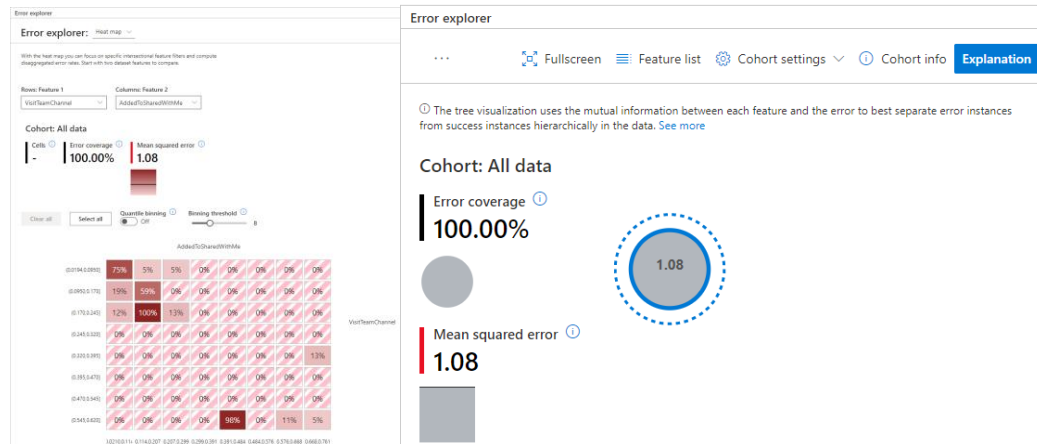


6. Run error analysis

- (1) Upload [error_analysis.ipynb](#) to AML and run it



Check the analysis result





II. Run your own experiment

The pipeline imported in the previous section is just template you can use to create your own machine learning pipeline. This section guides several ways to customize the template.

1. Explore data

Once new dataset is imported, you will need to manually explore data to find out features that could be useful for your model. The data created in the process is recommended to be saved in XXX.

2. Engineer features

2.1 Create featurization notebook

With the features found in the previous step, you can create a notebook to create intermediate dataset that contains features for each key. For example, if you need to create a machine learning model to identify vulnerable students, the intermediate dataset should have features associated with each student, such as the number of absences with Student ID. You can find sample notebook activities in the pipeline named “Engineer features in Stage2p”.

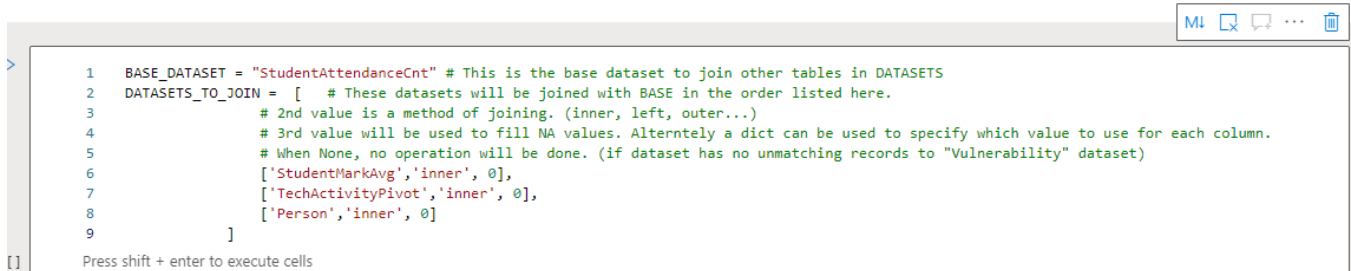
2.2 Include notebook into pipeline

Open the third activity named “Engineer features in Stage2p” in the pipeline, look at another pipeline named “FeatureEngineering”, and add a new “notebook” activity that executes the notebook you implemented in the previous step.

3. Re-create main dataset

You can include the intermediate dataset you created in the previous step into “Main” dataset by modifying Config section of “create_main_dataset” notebook in the pipeline activity named “Create main dataset in Stage3p”. This notebook joins datasets that are created in the previous step and saves it in Stage3p. You can make changes as described in the comments.

Configuration for data creation (for “create_main_dataset” notebook)



```
1 BASE_DATASET = "StudentAttendanceCnt" # This is the base dataset to join other tables in DATASETS
2 DATASETS_TO_JOIN = [ # These datasets will be joined with BASE in the order listed here.
3     # 2nd value is a method of joining. (inner, left, outer...)
4     # 3rd value will be used to fill NA values. Alternately a dict can be used to specify which value to use for each column.
5     # When None, no operation will be done. (if dataset has no unmatching records to "Vulnerability" dataset)
6     ['StudentMarkAvg','inner', 0],
7     ['TechActivityPivot','inner', 0],
8     ['Person','inner', 0]
9 ]
```

[] Press shift + enter to execute cells

4. Change dataset for training

Training datasets needs to be registered to AML before submitting an experiment. This can be done by editing “ml_config” notebook and running “register_dataset_to_aml” notebook. As described in 5.1, 3 datasets (“feature”, “label”, and “sensitive_feature”) are necessary to train a ML model and evaluate a trained ML model.

They are the subsets of main dataset created in the previous step. Names of columns needs to be listed as “key”, “feature” (or “exclude feature”), “sensitive_feature”, and “label”.

“sensitive_feature” is used to evaluate fairness after training a model.



For details of config elements, you can check the comments in “ml_config” notebook.

Configuration for AML

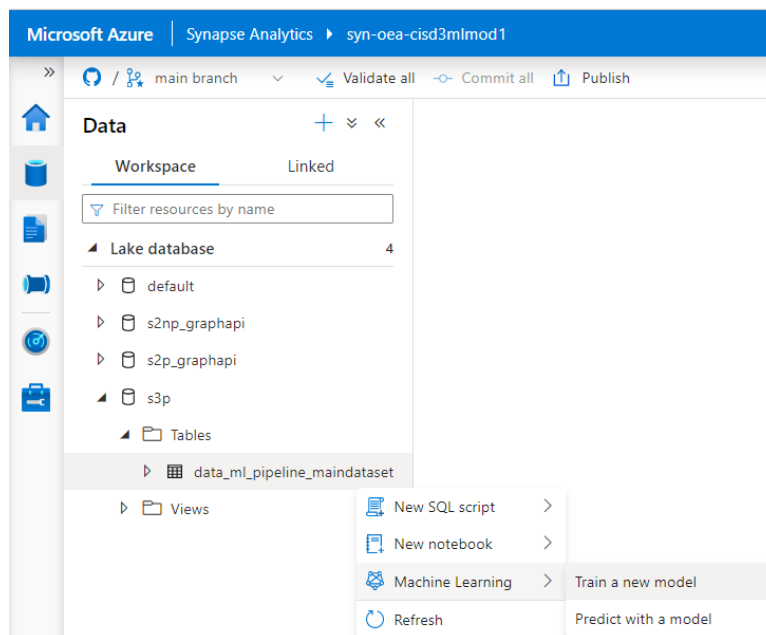
```
1 ML_CONFIG = {
2     "key": {"PersonId_pseudonym"},           # Key columns in dataset
3     "feature": None,                         # Feature columns in dataset (When None, all the features except label and exclude_feature will used)
4     "exclude_feature": None,                 # Columns in dataset to be excluded from features
5     "sensitive_feature": {"GenerationCode"}, # Sensitive feature columns in dataset (When None or {}, no dataset for sensitive features will be registered)
6     "label": {"avg_numeric_grade"},          # Label columns in dataset
7     "label_dtype": "double",                 # Data type of label
8     "model": "regression_linearreg",         # Model class name
9 }
10 MODEL_VERSION = '1'                        # Version of trained model used for prediction (Version can be checked in "Models" tab in AML)
11                                             # "predict" feature requires to specify version (it doesn't have feature to use the latest version)
12 LINKED_SERVICE_NAME = "LS_ML"              # Link Service from Synapse to AML
13 COMPUTE = "testcluster"                    # Compute in AML
14
15 URI = "https://github.com/lladop/OpenEduAnalytics.git#packages/How_to_implement_Azure_machine_learning/aml_modeling/project" # URI to the mlflow project in Github repo
16 BRANCH = "maidap-fy22h1-mlpipeline"        # Branch of Github repo
17
18 # When repo is private, URI needs to be set with personal access token as shown below
19 #URI = f"https://{{TOKEN}}@github.com/ContosoISD3/cisdggimpl5.git#aml_projects/mlflow/template_project/project/"
20 # Personal Access Token (No need to modify this for now.
21 # https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token)
22
23 Press shift + enter to execute cells
```

5. Submit an experiment to AML

Once training dataset is created, an experiment can be submitted to AML. In the example above, a linear regression model is used as a machine learning model. But there are several other models you can use without coding as shown in the class diagram included in “7.3 Implement model”. You can specify one of these as “model” in the config notebook above.

6. Run AutoML

Instead of running a model in the scripts saved in You can also run AutoML by clicking on “Data” tab > “data_ml_pipeline_maindataset” in s3p > “Machine Learning” > “Train a new model”.



You can find detailed documents below.



- [Tutorial: Train a model by using automated machine learning - Azure Synapse Analytics | Microsoft Docs](#)

7. Add a new model to AML

As described in “5.Submit an experiment to AML”, there are models already ready to use without coding. If it is necessary to use a model not on the list, it is also possible to create a new model by coding. By implementing a model with certain interface defined in the framework, you can check “Metrics”, “Explanations”, and “Fairness” in AML as you see in 5.2(2).

7.1 Setup development environment

- (1) You need Python development environment with [Anaconda](#).
- (2) Create virtual environment

Open CUI(*), go to “aml_code” folder, run the following command.

```
conda env create -f dev-env.yaml
```

(*) e.g. windows: Anaconda Prompt, mac: Terminal

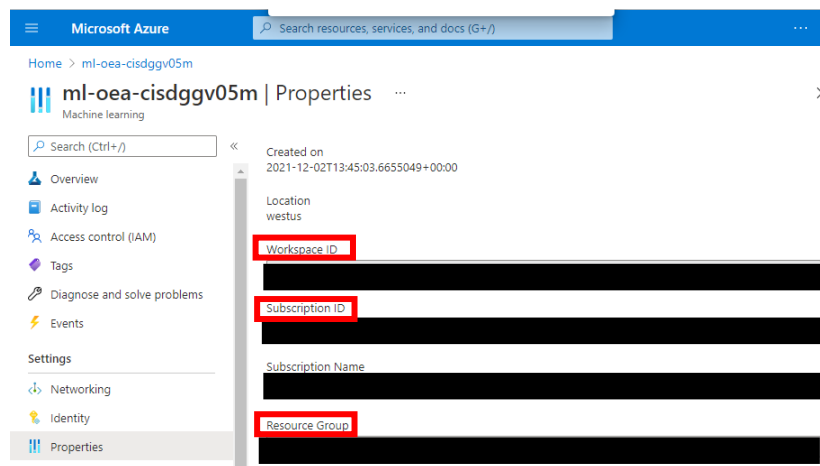
- (3) Activate virtual environment

Run the following command.

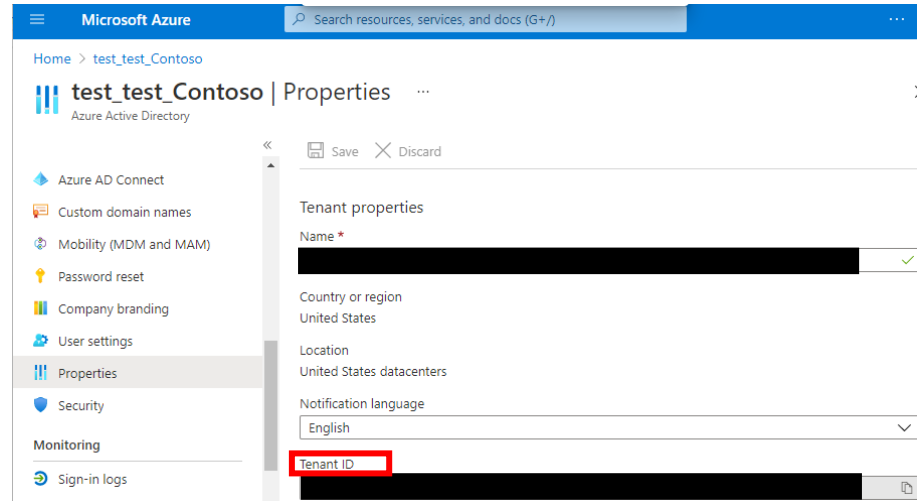
```
conda activate dev-oea-modeling
```

7.2 Submit experiment with default configuration

- (1) Download the code from [OEA Github repository](#)
- (2) Copy “sample_config.json” as “config.json” and modify it
You can find your environment variables in Azure portal as shown below. “compute” is the one created in 2.1(4).
 - Azure portal > AML Instance



- Azure Portal > Azure Active Directory



- (3) Move to the one with “submit_pipelines.py” and run

```
python submit_pipeline.py
```

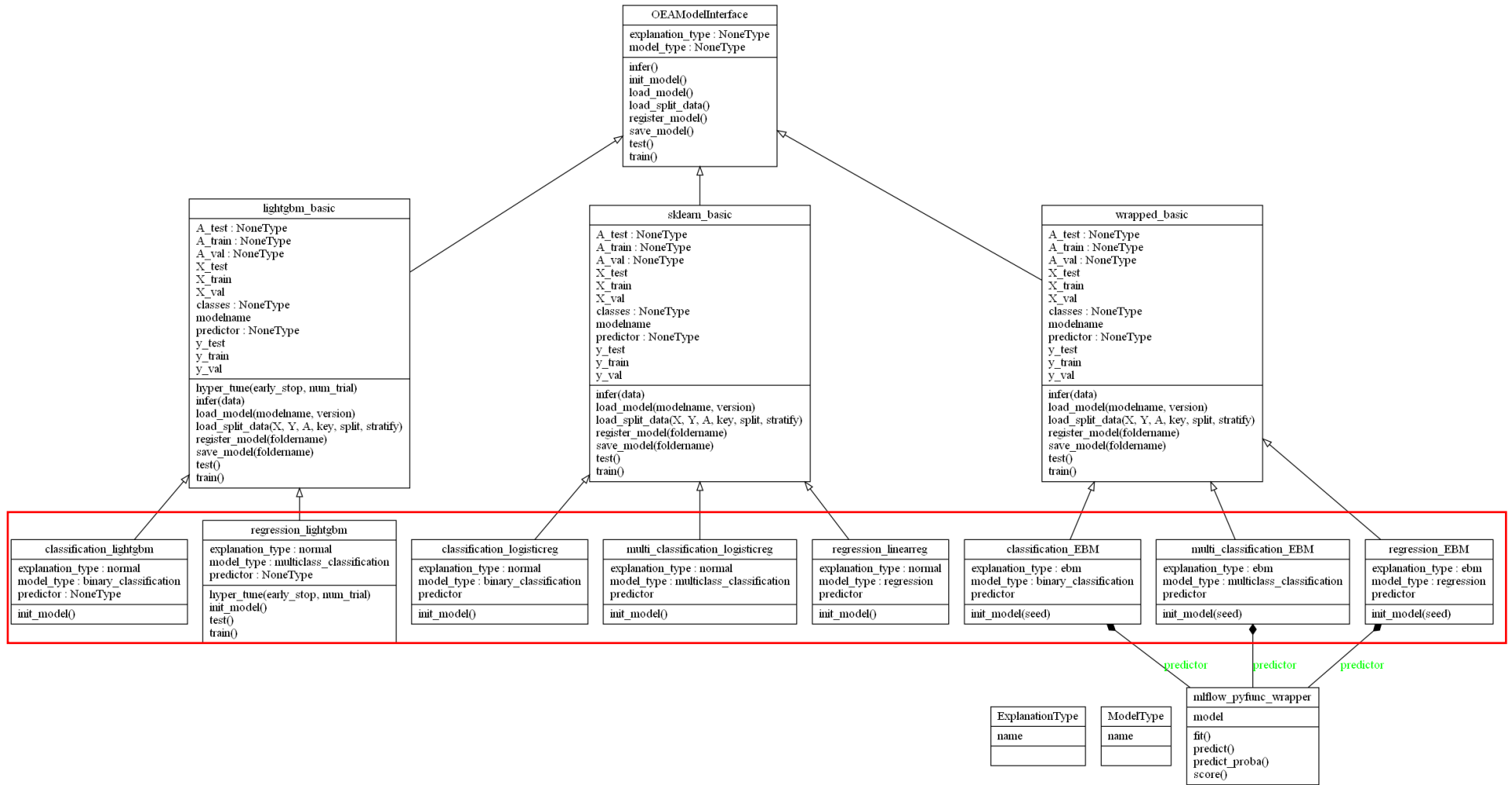
- (4) Click “Experiments” tab > “template_test” on AML and make sure that there is one submitted by you.

Display name	Status	Submitted time ↓	Duration	Submitted by	Compute target	Run type	Last(training_s...	Last(training...
polite_quill_qhd2lsz9	Completed	Nov 30, 2021 1:57 PM	1m 44s		mlcompute1117	Script	0.273	0.136

7.3 Implement model

- (1) Create a new python file in “aml_code/modeling” folder
- (2) Define a new model class in the new python file. This class needs to implement methods and specify class variables (“model_type” and “explanation_type”) that are defined in OEA_model.py > “OEAModelInterface” class.

You can check several model classes included in the repository as a reference. The following class diagram shows those classes and their relationships.





- (3) Modify the following variables in “submit_pipelines.py”. “model” needs to be changed to the name of the model class you created. If you have registered new datasets to AML in “4. Change dataset for training”, it is also necessary to change “feature_dataset_name”, “label_dataset_name”, “sensitive_dataset_name”, and “key”.

```
def submit_job(ws, experiment_name, entry="main", compute="DEFAULT_COMPUTE_REPLACE"):  
  
    backend_config = {"COMPUTE": compute, "USE_CONDA": True}  
  
    backend_flag = "azureml"  
    datastore = "datastore_stage3"  
    feature_dataset_name = 'dataset-dataconfigA-feature'  
    label_dataset_name = 'dataset-dataconfigA-label'  
    sensitive_dataset_name = 'dataset-dataconfigA-sensitive_feature'  
    key = '["StudentDwRefId_pseudonym"]'  
    model = 'classification_logisticreg'  
    registered_name = 'model_Azure_In_Action'
```

- (4) Submit an experiment by running the same command used in 7.2(3). Click “Experiments” tab > “template_test” on AML and make sure that there is one submitted by you.


7.4 Train the new model from Azure Synapse

- (1) In Synapse, change “model” in “ml_config” notebook to the name of the model class created in the previous step.

Configuration for AML

```
1  ML_CONFIG = {  
2      "key": ("PersonId_pseudonym"),          # Key columns in dataset  
3      "feature": None,                        # Feature columns in dataset (when None, all the features except label and exclude_feature will used)  
4      "exclude_feature": None,                # Columns in dataset to be excluded from features  
5      "sensitive_feature": ("GenerationCode"), # Sensitive feature columns in dataset (when None or [], no dataset for sensitive features will be registered)  
6      "label": ("avg_numeric_grade"),          # Label columns in dataset  
7      "label_type": "numeric",                 # Data type of label  
8      "model": "regression_linearreg",         # Model class name  
9  }  
10 MODEL_VERSION = '1'                         # Version of trained model used for prediction (Version can be checked in "Models" tab in AML)  
11 # "predict" feature requires to specify version (it doesn't have feature to use the latest version)  
12 LINKED_SERVICE_NAME = "LS_ML"               # Link service from Synapse to AML  
13 COMPUTE = "testcluster"                     # Compute in AML  
14  
15 URI = "https://github.com/lladap/OpenEduAnalytics.git#packages/How_to_implement_Azure_machine_learning/aml_modeling/project" # URI to the mlflow project in Github repo  
16 BRANCH = "msidap-fy22h1-mlpipeline"         # Branch of Github repo  
17  
18 # When repo is private, URI needs to be set with personal access token as shown below  
19 #URI = f"https://(TOKEN)@github.com/ContosoIS03/cisdgimpl5.git#aml_projects/mlflow/template_project/project/"  
20 # Personal Access Token (No need to modify this for now.)  
21 # https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token  
Press shift + enter to execute cells
```

- (2) Run “submit_mlflow_experiment_to_aml” notebook. Go to AML, click “Experiments” tab > “template_test”, and make sure that there is one submitted by “Service Principal” and completed.

Display name	Status	Submitted time ↓	Duration	Submitted by
 polite_quill_qhd2lsz9 nked Services	✓ Completed	Nov 30, 2021 1:57 PM	1m 44s	Service Principal

- (3) Run the whole pipeline and make sure everything works fine