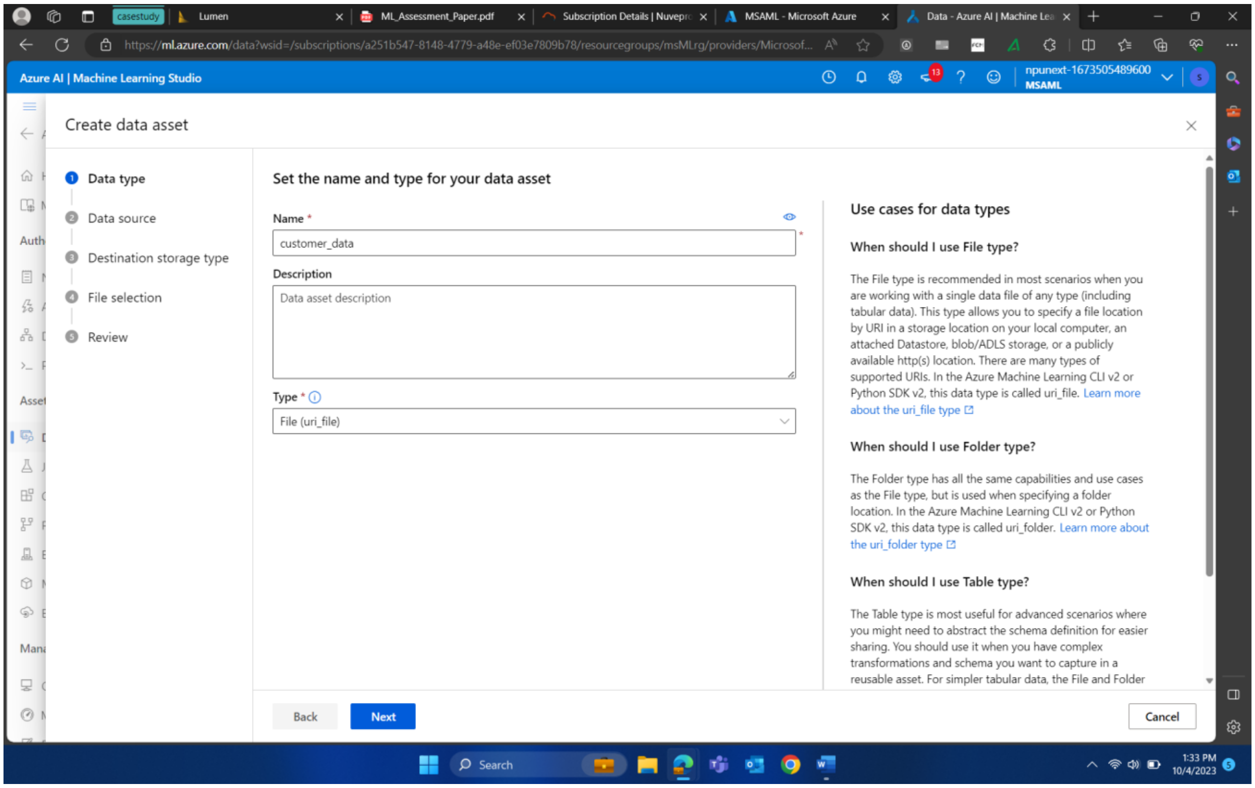
Megha Sharma

Megha.sharma2@shell.com

Hands On assignment

1 a.


casestudy
Lumen
ML_Assessment_Paper.pdf
Subscription Details | Nuvepro
MSAML - Microsoft Azure
Data - Azure AI | Machine Lea
https://ml.azure.com/data?wsid=/subscriptions/a251b547-8148-4779-a48e-ef03e7809b78/resourcegroups/msMLrg/providers/Microsof...
Azure AI | Machine Learning Studio
13
npunext-1673505489600
MSAML
Search
1:33 PM
10/4/2023

Create data asset

1 Data type
2 Data source
3 Destination storage type
4 File selection
5 Review

Set the name and type for your data asset

Name *
customer_data

Description
Data asset description

Type *
File (uri_file)

Use cases for data types

When should I use File type?

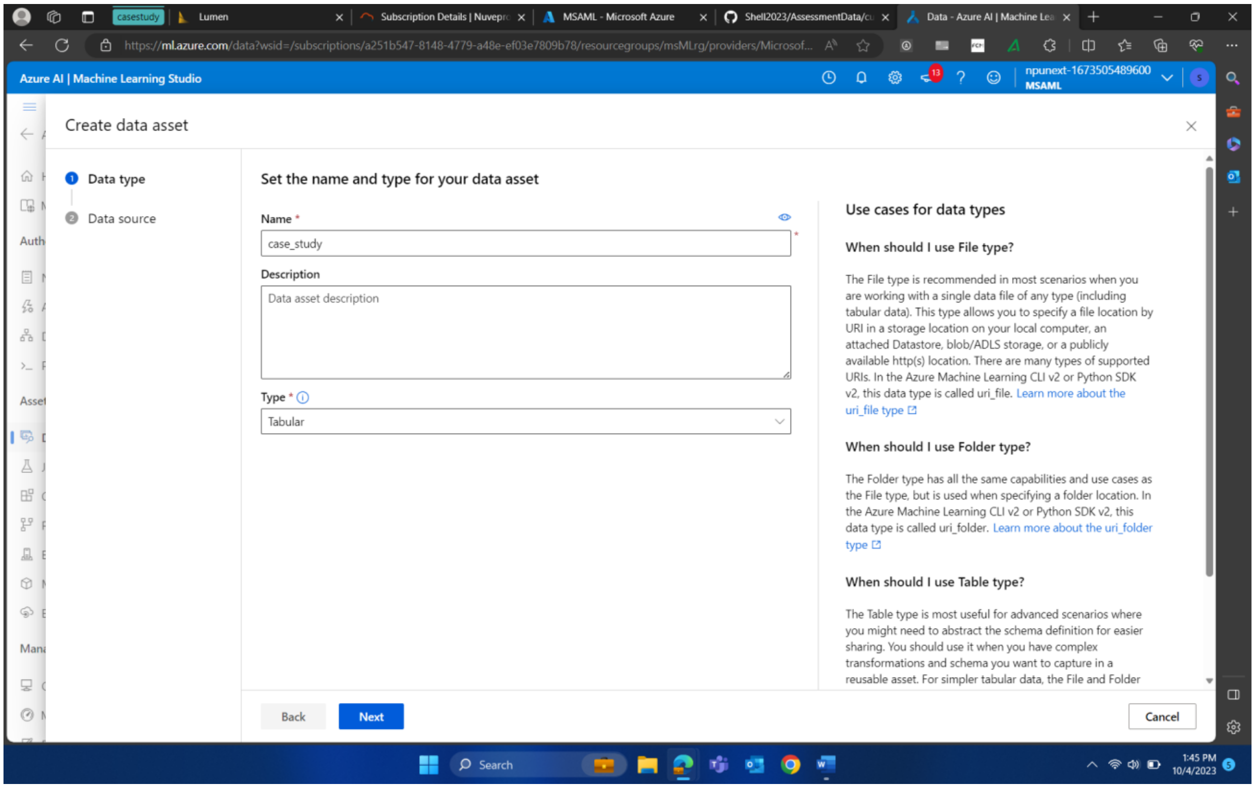
The File type is recommended in most scenarios when you are working with a single data file of any type (including tabular data). This type allows you to specify a file location by URI in a storage location on your local computer, an attached Datastore, blob/ADLS storage, or a publicly available http(s) location. There are many types of supported URIs. In the Azure Machine Learning CLI v2 or Python SDK v2, this data type is called uri_file. Learn more about the uri_file type

When should I use Folder type?

The Folder type has all the same capabilities and use cases as the File type, but is used when specifying a folder location. In the Azure Machine Learning CLI v2 or Python SDK v2, this data type is called uri_folder. Learn more about the uri_folder type

When should I use Table type?

The Table type is most useful for advanced scenarios where you might need to abstract the schema definition for easier sharing. You should use it when you have complex transformations and schema you want to capture in a reusable asset. For simpler tabular data, the File and Folder



Azure AI | Machine Learning Studio

Create data asset

1. Data type
2. Data source

Set the name and type for your data asset

Name *

case_study

Description

Data asset description

Type *

Tabular

Use cases for data types

When should I use File type?

The File type is recommended in most scenarios when you are working with a single data file of any type (including tabular data). This type allows you to specify a file location by URI in a storage location on your local computer, an attached Datastore, blob/ADLS storage, or a publicly available http(s) location. There are many types of supported URIs. In the Azure Machine Learning CLI v2 or Python SDK v2, this data type is called uri_file. Learn more about the uri_file type

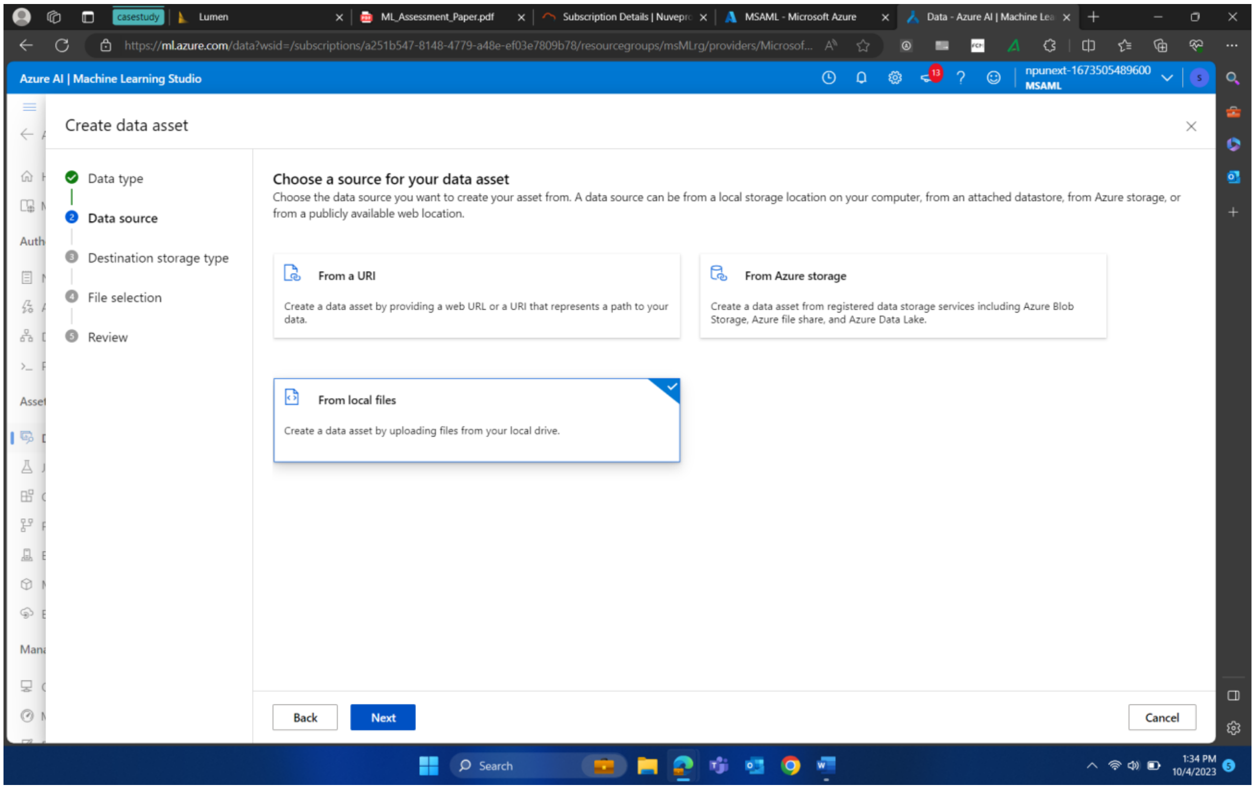
When should I use Folder type?

The Folder type has all the same capabilities and use cases as the File type, but is used when specifying a folder location. In the Azure Machine Learning CLI v2 or Python SDK v2, this data type is called uri_folder. Learn more about the uri_folder type

When should I use Table type?

The Table type is most useful for advanced scenarios where you might need to abstract the schema definition for easier sharing. You should use it when you have complex transformations and schema you want to capture in a reusable asset. For simpler tabular data, the File and Folder

Data type



Azure AI | Machine Learning Studio

Create data asset

- Data type
2. Data source
3. Destination storage type
4. File selection
5. Review

Choose a source for your data asset

Choose the data source you want to create your asset from. A data source can be from a local storage location on your computer, from an attached datastore, from Azure storage, or from a publicly available web location.

From a URI

Create a data asset by providing a web URL or a URI that represents a path to your data.

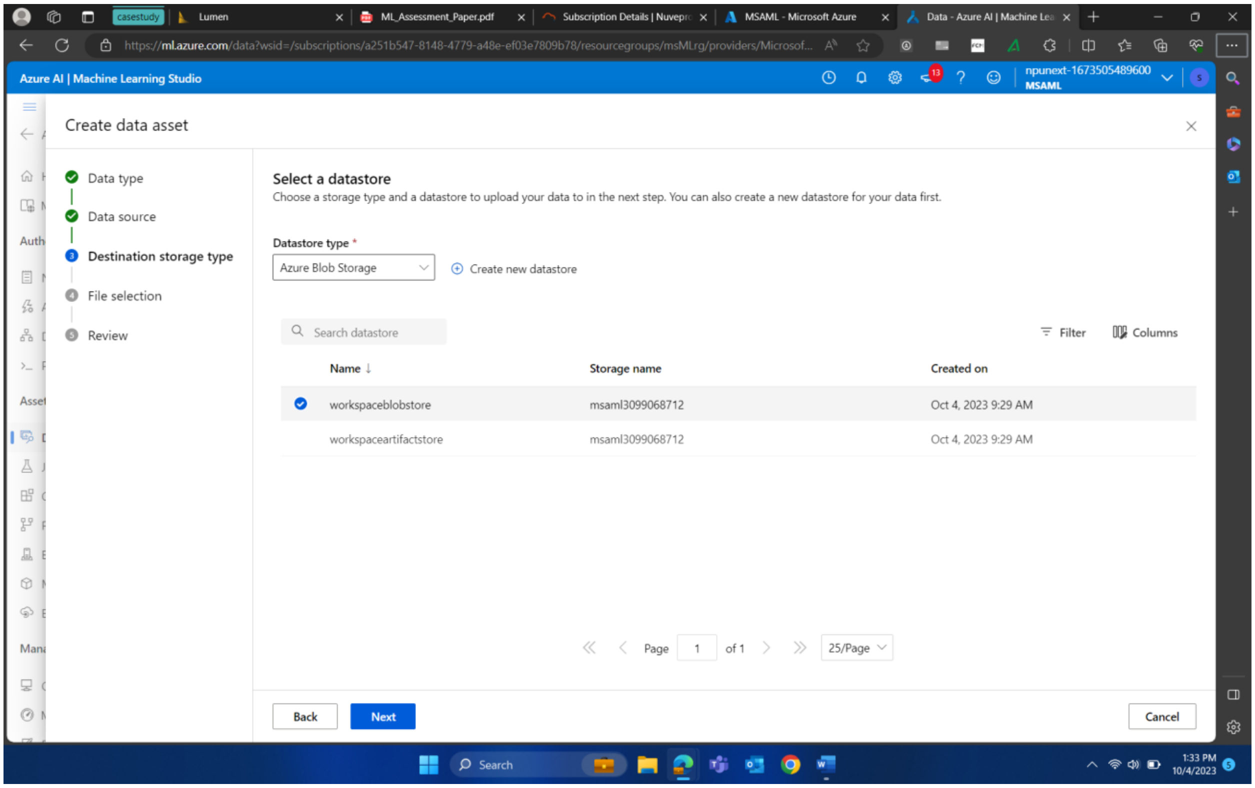
From Azure storage

Create a data asset from registered data storage services including Azure Blob Storage, Azure file share, and Azure Data Lake.

From local files ✓

Create a data asset by uploading files from your local drive.

Data source



Create data asset

- Data type
- Data source
- Destination storage type
- File selection
- Review

Select a datastore

Choose a storage type and a datastore to upload your data to in the next step. You can also create a new datastore for your data first.

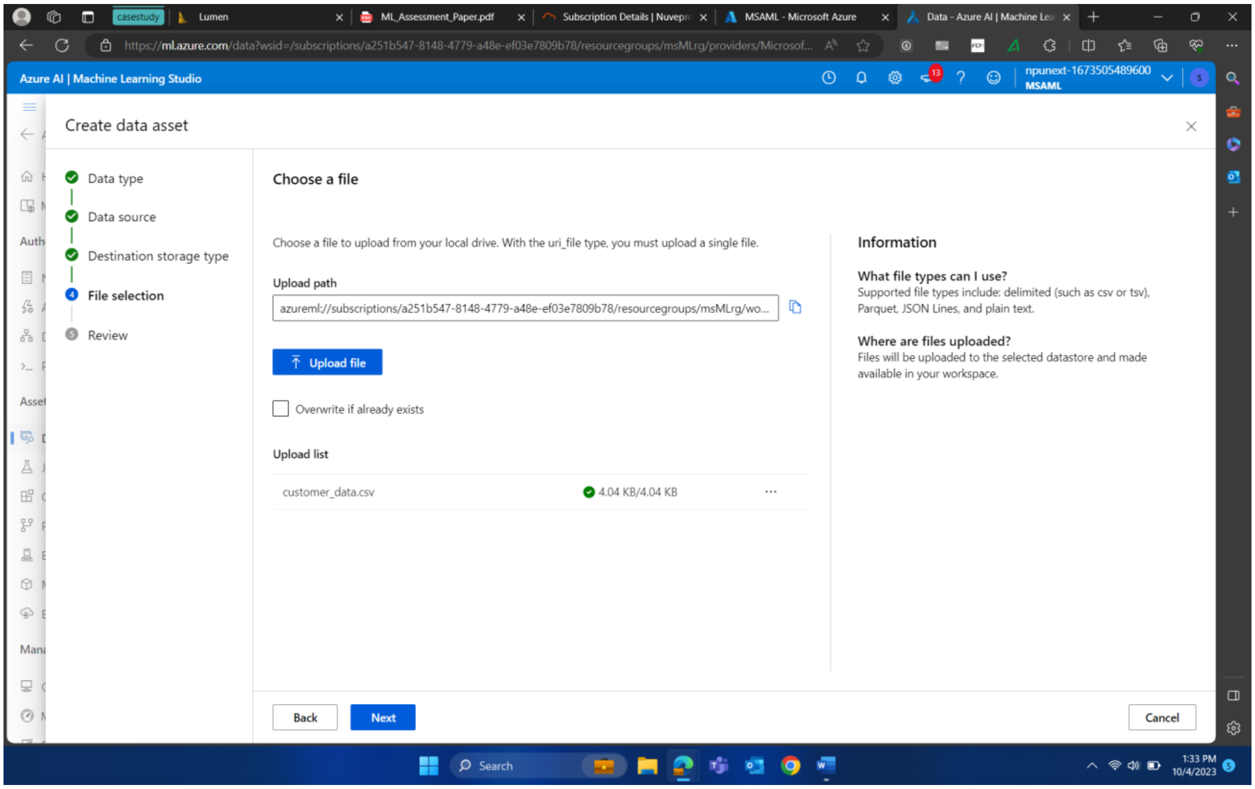
Datastore type *

Azure Blob Storage

Create new datastore

Name ↓	Storage name	Created on
workspaceblobstore	msaml3099068712	Oct 4, 2023 9:29 AM
workspaceartifactstore	msaml3099068712	Oct 4, 2023 9:29 AM

Blob storage destination



Create data asset

- Data type
- Data source
- Destination storage type
- File selection
- Review

Choose a file

Choose a file to upload from your local drive. With the uri_file type, you must upload a single file.

Upload path

azureml://subscriptions/a251b547-8148-4779-a48e-ef03e7809b78/resourcegroups/msMLrg/wo...

Upload file

□ Overwrite if already exists

Upload list

customer_data.csv ✓ 4.04 KB/4.04 KB

Information

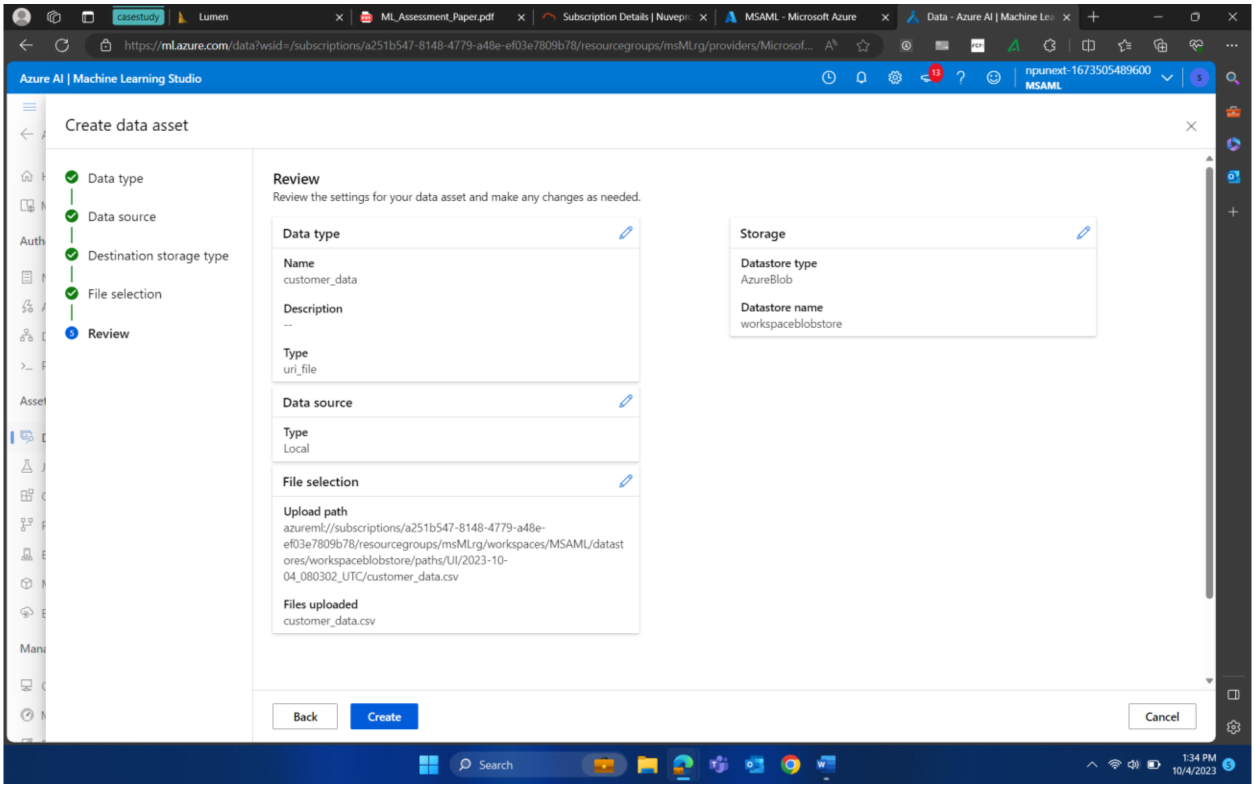
What file types can I use?

Supported file types include: delimited (such as csv or tsv), Parquet, JSON Lines, and plain text.

Where are files uploaded?

Files will be uploaded to the selected datastore and made available in your workspace.

File selection



Create data asset

- ✓ Data type
- ✓ Data source
- ✓ Destination storage type
- ✓ File selection
- 5 Review

Review

Review the settings for your data asset and make any changes as needed.

Data type

Name
customer_data

Description
--

Type
uri_file

Data source

Type
Local

File selection

Upload path
azureml://subscriptions/a251b547-8148-4779-a48e-ef03e7809b78/resourcegroups/msMLrg/workspaces/MSAML/datastores/workspaceblobstore/paths/UI/2023-10-04_080302_UTC/customer_data.csv

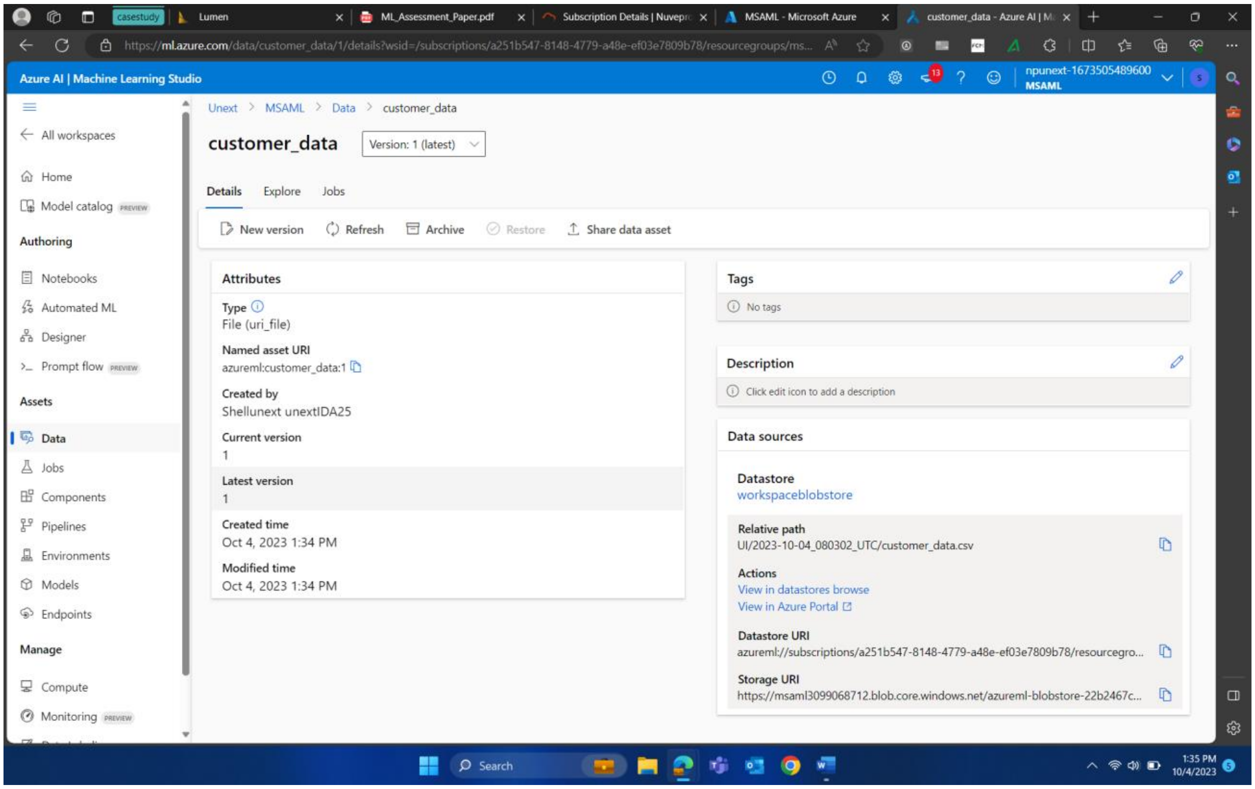
Files uploaded
customer_data.csv

Storage

Datastore type
AzureBlob

Datastore name
workspaceblobstore

Review data asset



customer_data Version: 1 (latest)

Attributes

Type
File (uri_file)

Named asset URI
azureml:customer_data:1

Created by
Shellunext unextIDA25

Current version
1

Latest version
1

Created time
Oct 4, 2023 1:34 PM

Modified time
Oct 4, 2023 1:34 PM

Tags

No tags

Description

Click edit icon to add a description

Data sources

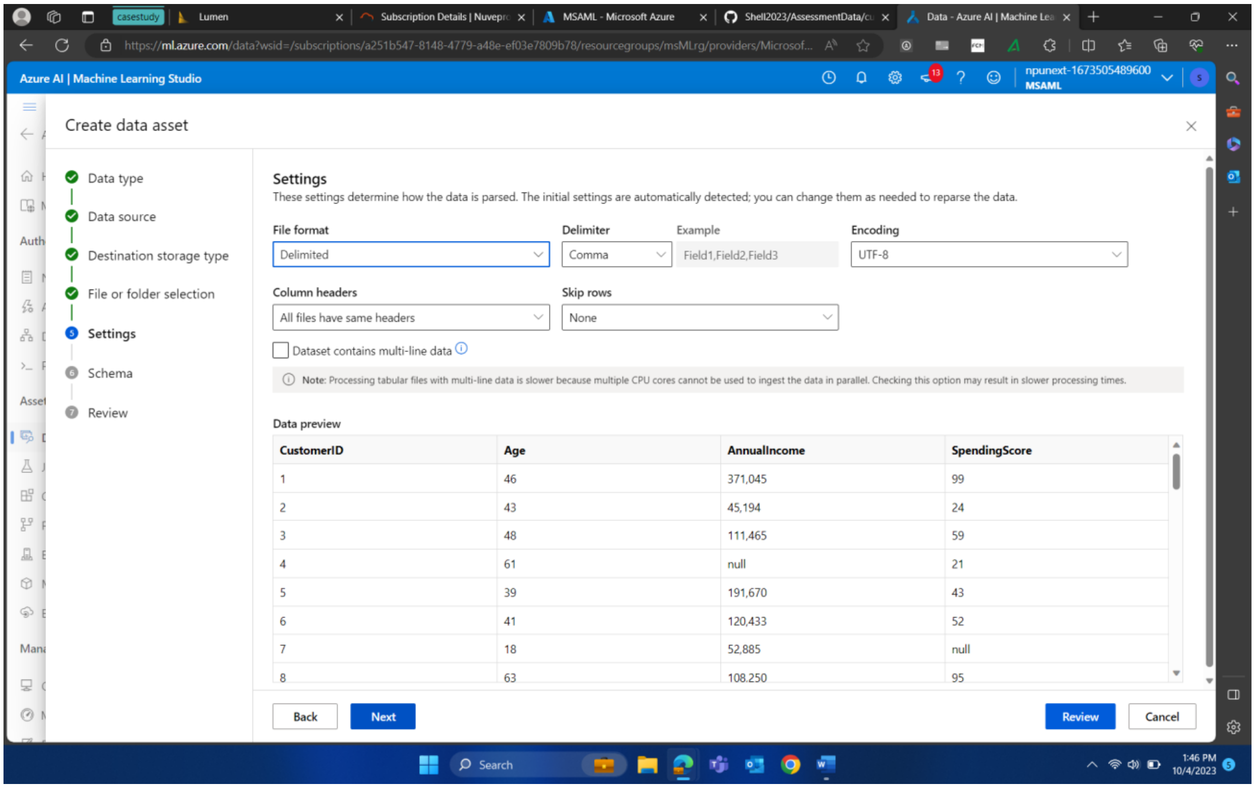
Datastore
workspaceblobstore

Relative path
UI/2023-10-04_080302_UTC/customer_data.csv

Actions
View in datastores browse
View in Azure Portal

Datastore URI
azureml://subscriptions/a251b547-8148-4779-a48e-ef03e7809b78/resourcegro...

Storage URI
https://msaml3099068712.blob.core.windows.net/azureml-blobstore-22b2467c...


casestudy
Lumen
Subscription Details | Nuvepro
MSAML - Microsoft Azure
Shell2023/AssessmentData/cu
Data - Azure AI | Machine Lea
https://ml.azure.com/data?wsid=/subscriptions/a251b547-8148-4779-a48e-ef03e7809b78/resourcegroups/msMLrg/providers/Microsof...
Azure AI | Machine Learning Studio
npunext-1673505489600
MSAML
Search
1:46 PM
10/4/2023

Create data asset

- Data type
- Data source
- Destination storage type
- File or folder selection
- 5 Settings
- 6 Schema
- 7 Review

Settings

These settings determine how the data is parsed. The initial settings are automatically detected; you can change them as needed to reparse the data.

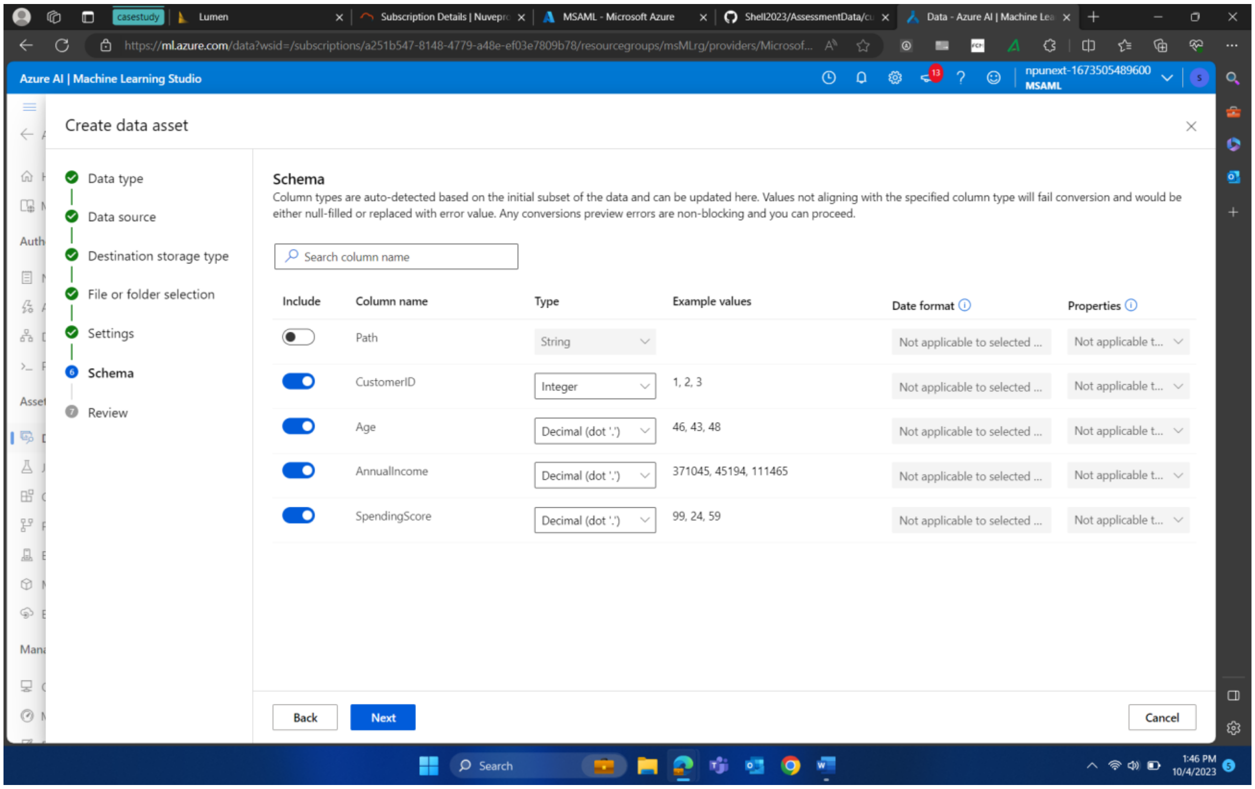
File format: Delimited
Delimiter: Comma
Example: Field1,Field2,Field3
Encoding: UTF-8
Column headers: All files have same headers
Skip rows: None

□ Dataset contains multi-line data

Note: Processing tabular files with multi-line data is slower because multiple CPU cores cannot be used to ingest the data in parallel. Checking this option may result in slower processing times.

Data preview

CustomerID	Age	AnnualIncome	SpendingScore
1	46	371,045	99
2	43	45,194	24
3	48	111,465	59
4	61	null	21
5	39	191,670	43
6	41	120,433	52
7	18	52,885	null
8	63	108,250	95


casestudy
Lumen
Subscription Details | Nuvepro
MSAML - Microsoft Azure
Shell2023/AssessmentData/cu
Data - Azure AI | Machine Lea
https://ml.azure.com/data?wsid=/subscriptions/a251b547-8148-4779-a48e-ef03e7809b78/resourcegroups/msMLrg/providers/Microsof...
Azure AI | Machine Learning Studio
npunext-1673505489600
MSAML
Search
1:46 PM
10/4/2023

Create data asset

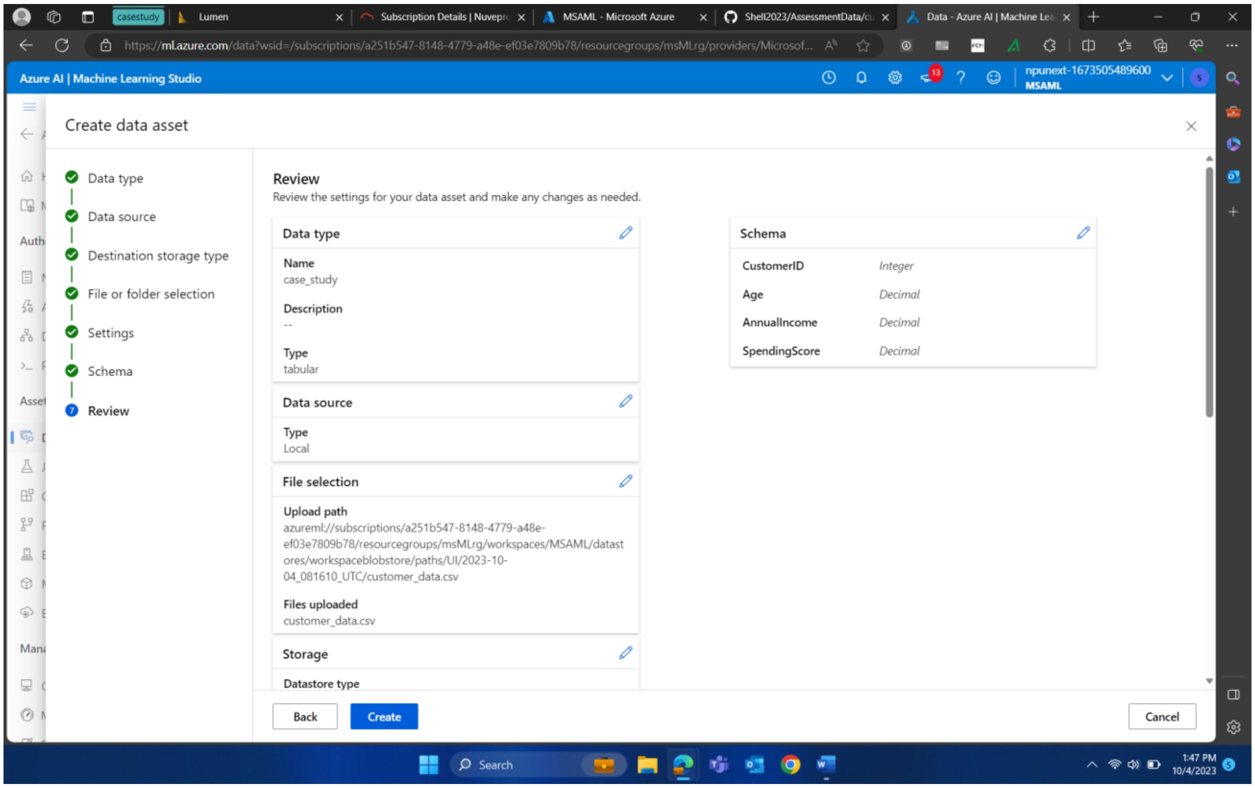
- Data type
- Data source
- Destination storage type
- File or folder selection
- Settings
- 6 Schema
- 7 Review

Schema

Column types are auto-detected based on the initial subset of the data and can be updated here. Values not aligning with the specified column type will fail conversion and would be either null-filled or replaced with error value. Any conversions preview errors are non-blocking and you can proceed.

Search column name

Include	Column name	Type	Example values	Date format	Properties
Off	Path	String		Not applicable to selected ...	Not applicable t...
On	CustomerID	Integer	1, 2, 3	Not applicable to selected ...	Not applicable t...
On	Age	Decimal (dot '.')	46, 43, 48	Not applicable to selected ...	Not applicable t...
On	AnnualIncome	Decimal (dot '.')	371045, 45194, 111465	Not applicable to selected ...	Not applicable t...
On	SpendingScore	Decimal (dot '.')	99, 24, 59	Not applicable to selected ...	Not applicable t...



Create data asset

Review

Review the settings for your data asset and make any changes as needed.

Data type

Name
case_study

Description
--

Type
tabular

Data source

Type
Local

File selection

Upload path
azureml://subscriptions/a251b547-8148-4779-a48e-ef03e7809b78/resourcegroups/msMLrg/workspaces/MSAML/datastores/workspaceblobstore/paths/UI/2023-10-04_081610_UTC/customer_data.csv

Files uploaded
customer_data.csv

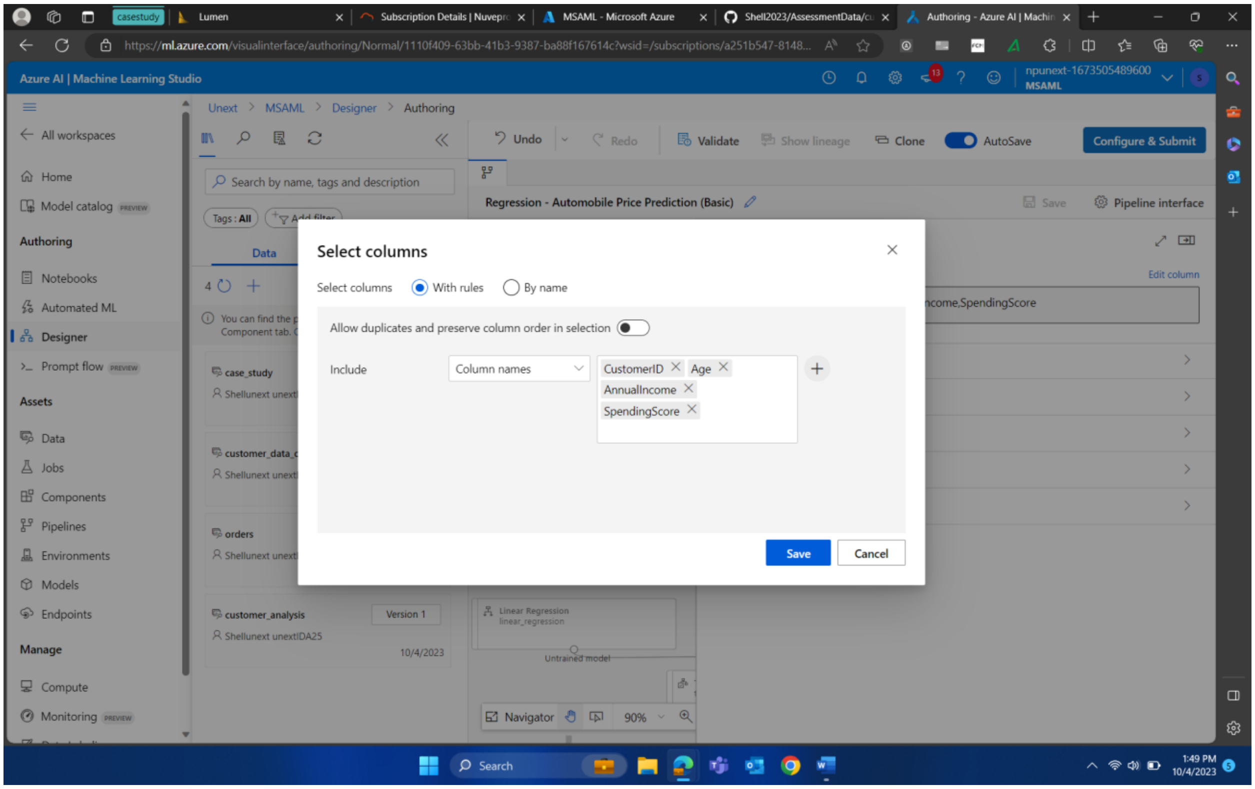
Storage

Datastore type

Schema

CustomerID	*Integer*
Age	*Decimal*
AnnualIncome	*Decimal*
SpendingScore	*Decimal*

1 b.



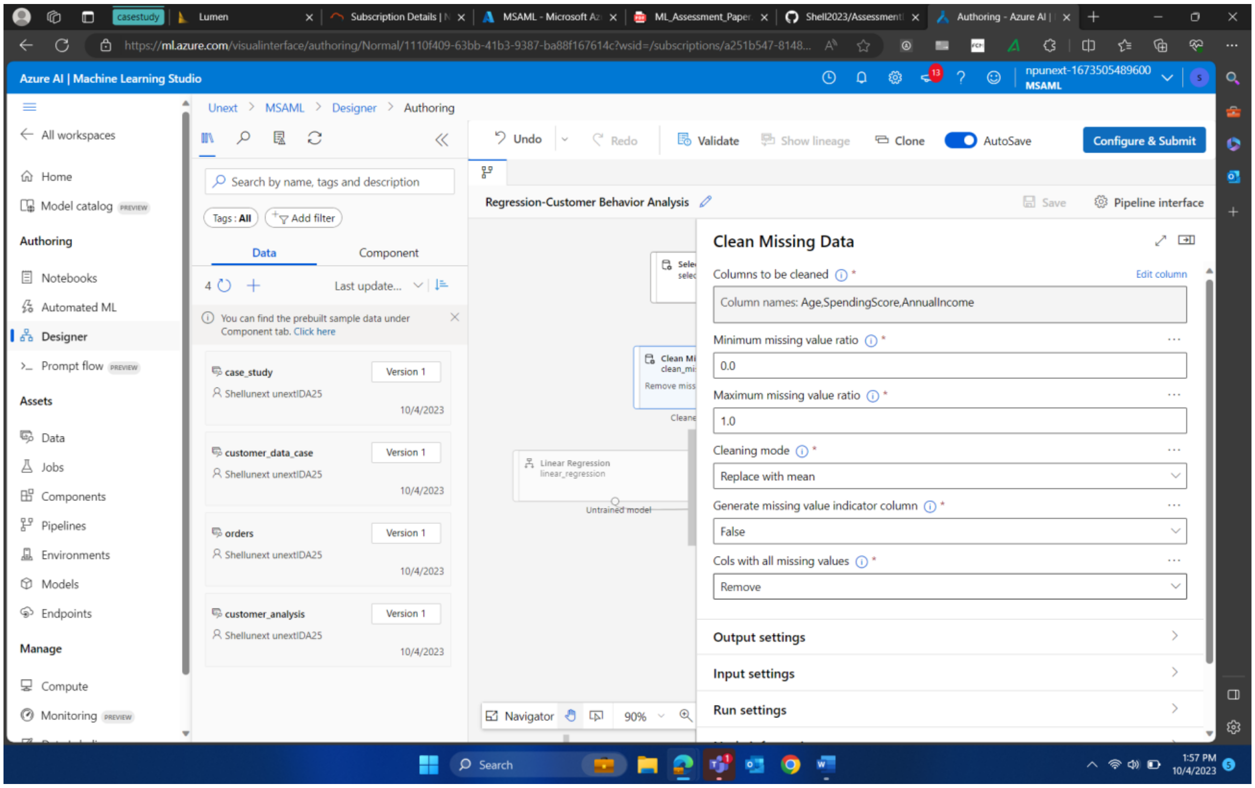
Select columns

Select columns ● With rules ○ By name

Allow duplicates and preserve column order in selection

Include | Column names | CustomerID, Age, AnnualIncome, SpendingScore

Select columns



Clean Missing Data

Columns to be cleaned *

Column names: Age,SpendingScore,AnnualIncome

Minimum missing value ratio *

0.0

Maximum missing value ratio *

1.0

Cleaning mode *

Replace with mean

Generate missing value indicator column *

False

Cols with all missing values *

Remove

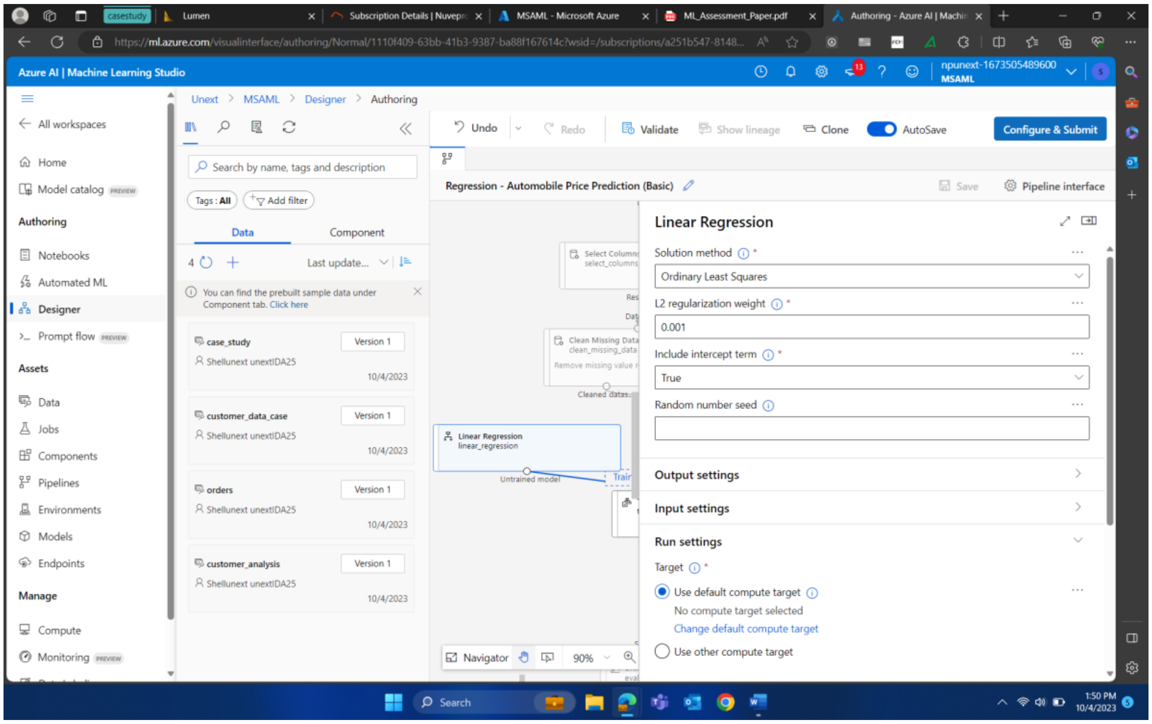
Output settings

Input settings

Run settings

Clean missing data

2 a.



Linear Regression

Solution method *

Ordinary Least Squares

L2 regularization weight *

0.001

Include intercept term *

True

Random number seed

Output settings

Input settings

Run settings

Target *

● Use default compute target

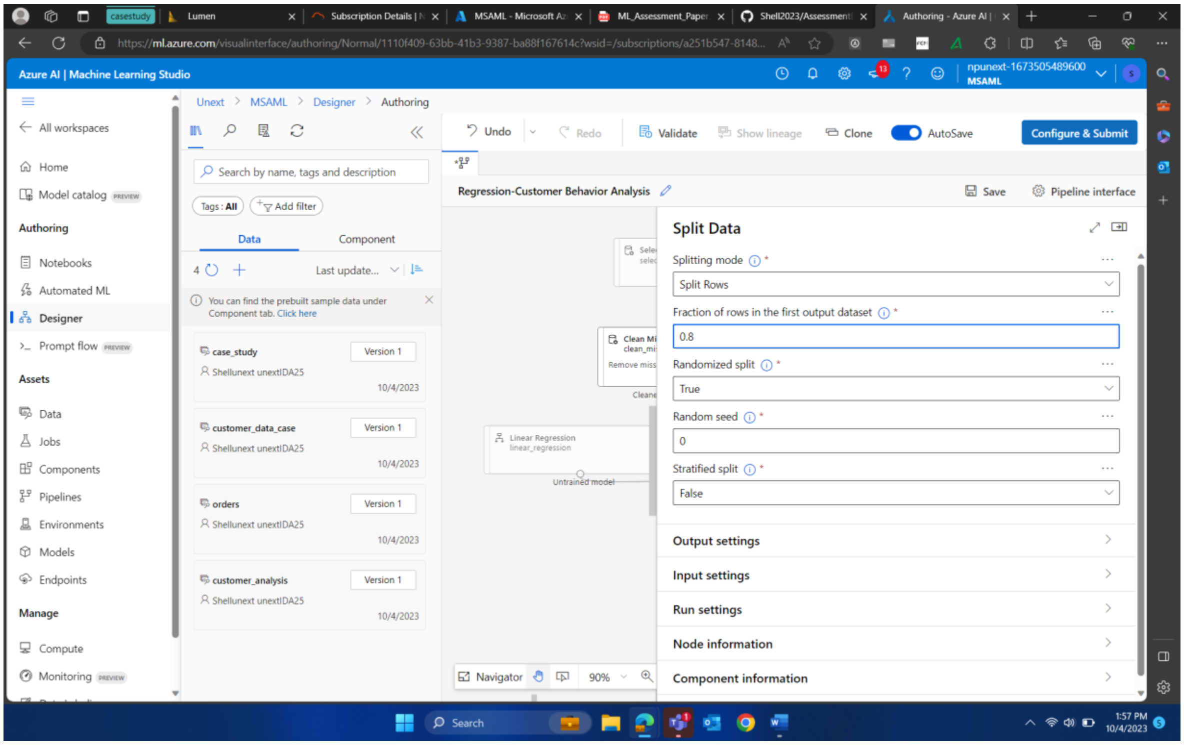
No compute target selected

Change default compute target

○ Use other compute target

Linear regression model

2 b.



Regression-Customer Behavior Analysis

Split Data

Splitting mode *
Split Rows

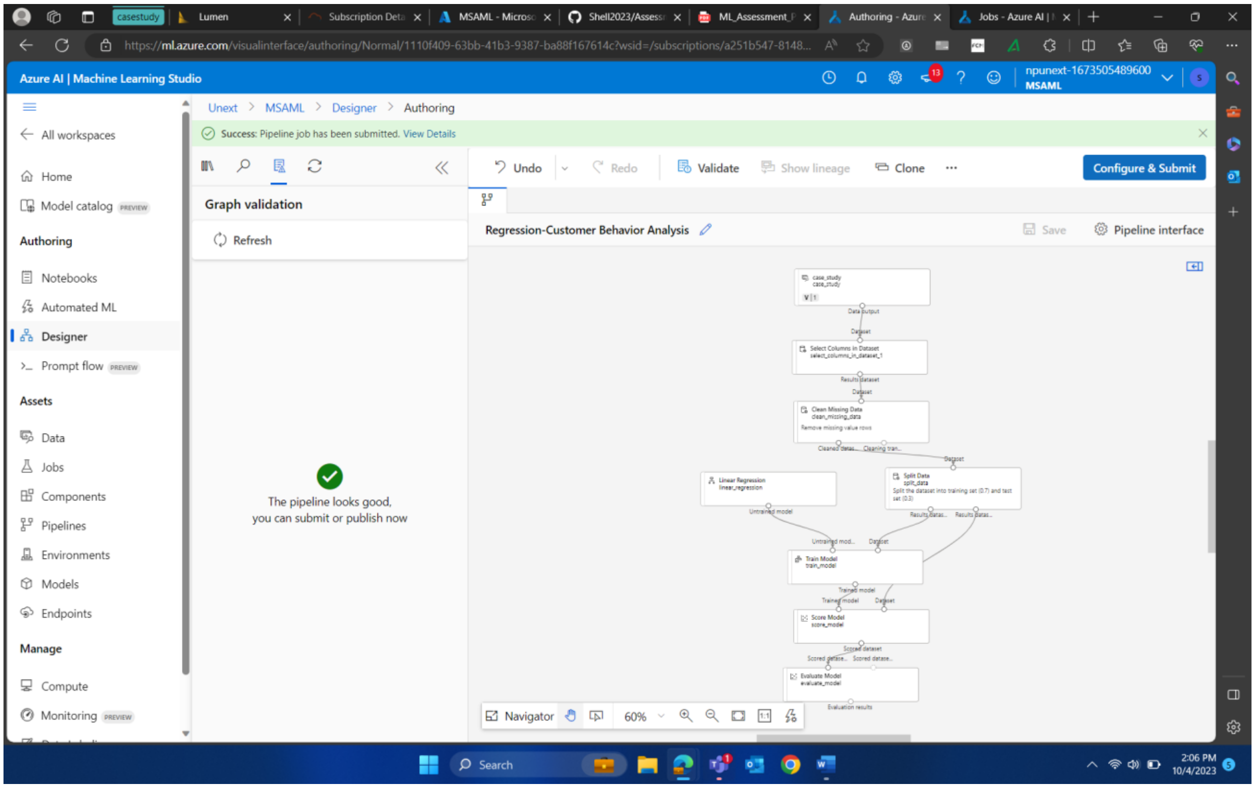
Fraction of rows in the first output dataset *
0.8

Randomized split *
True

Random seed *
0

Stratified split *
False

Split data



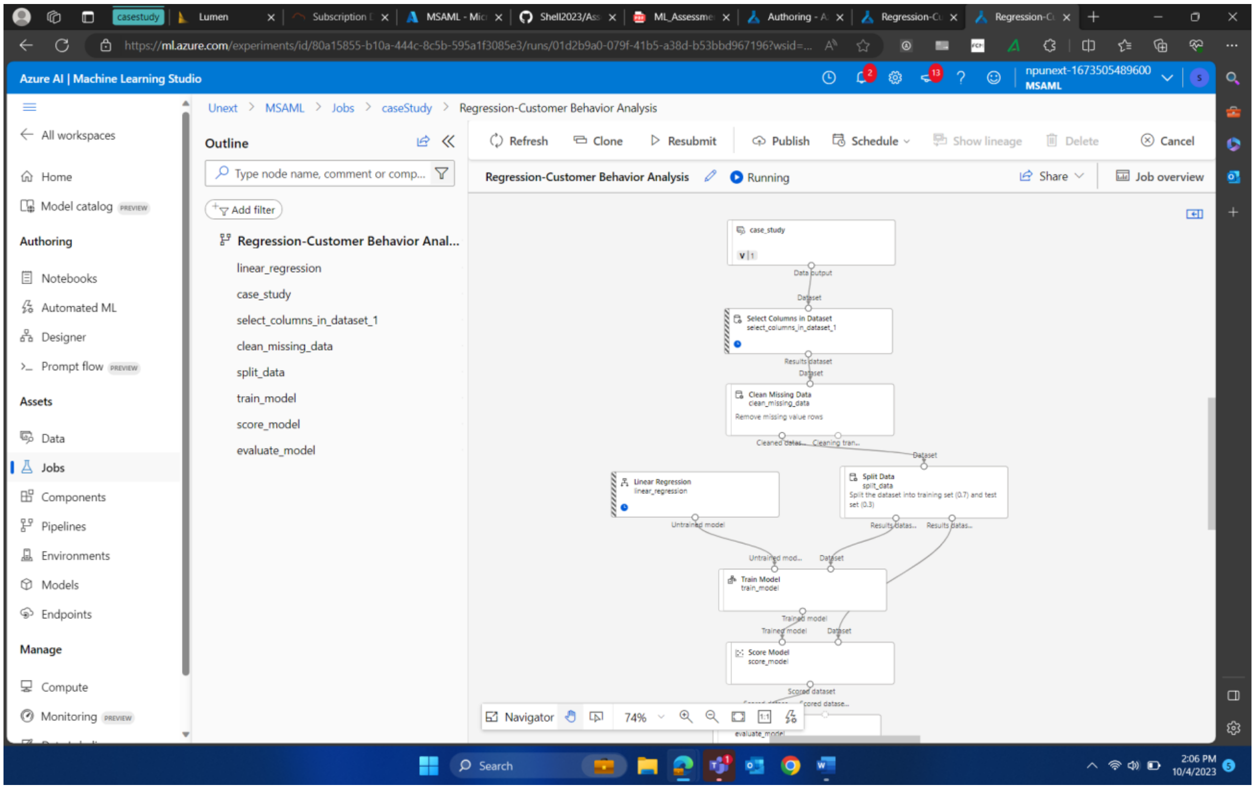
https://ml.azure.com/visualinterface/authoring/Normal/1110f409-63bb-41b3-9387-ba88f167614c?wsid=/subscriptions/a251b547-8148...

Success: Pipeline job has been submitted. View Details

Graph validation

The pipeline looks good, you can submit or publish now

Regression-Customer Behavior Analysis



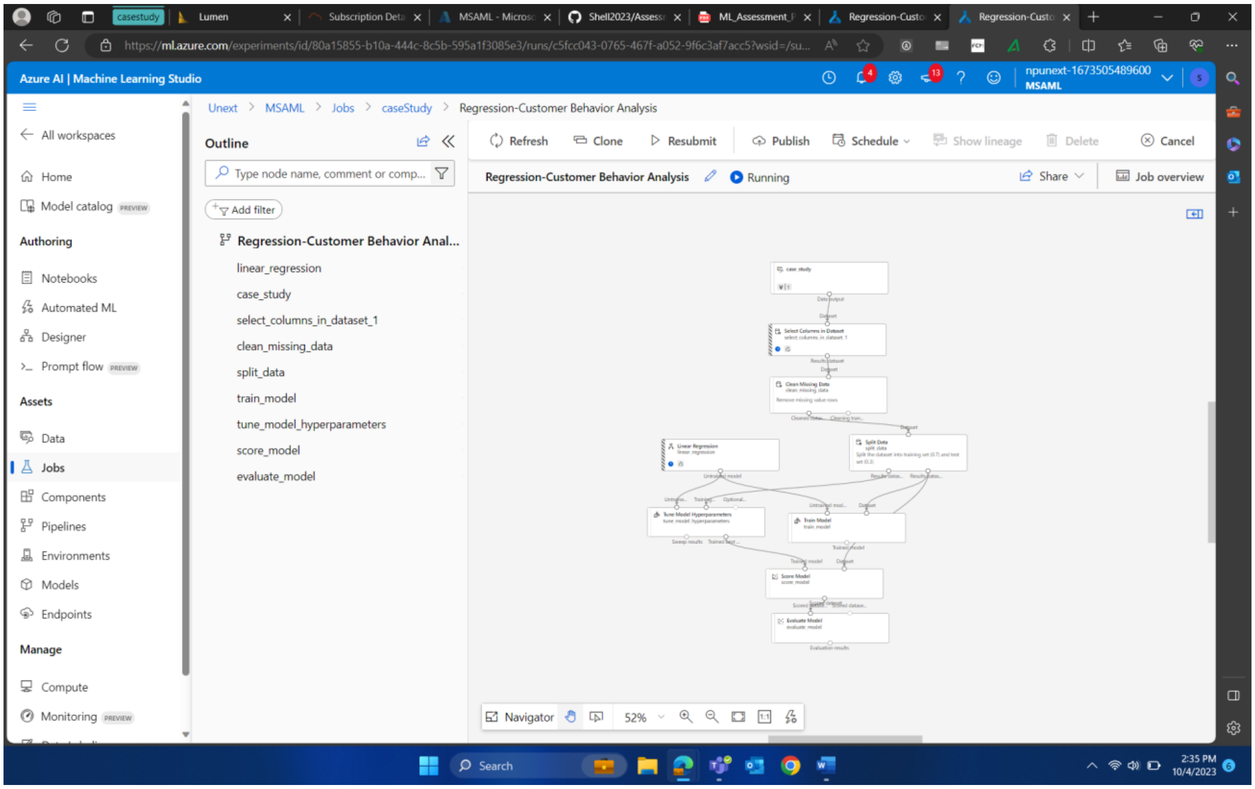
https://ml.azure.com/experiments/id/80a15855-b10a-444c-8c5b-595a1f3085e3/runs/01d2b9a0-079f-41b5-a38d-b53bbd967196?wsid=...

Regression-Customer Behavior Analysis — Running

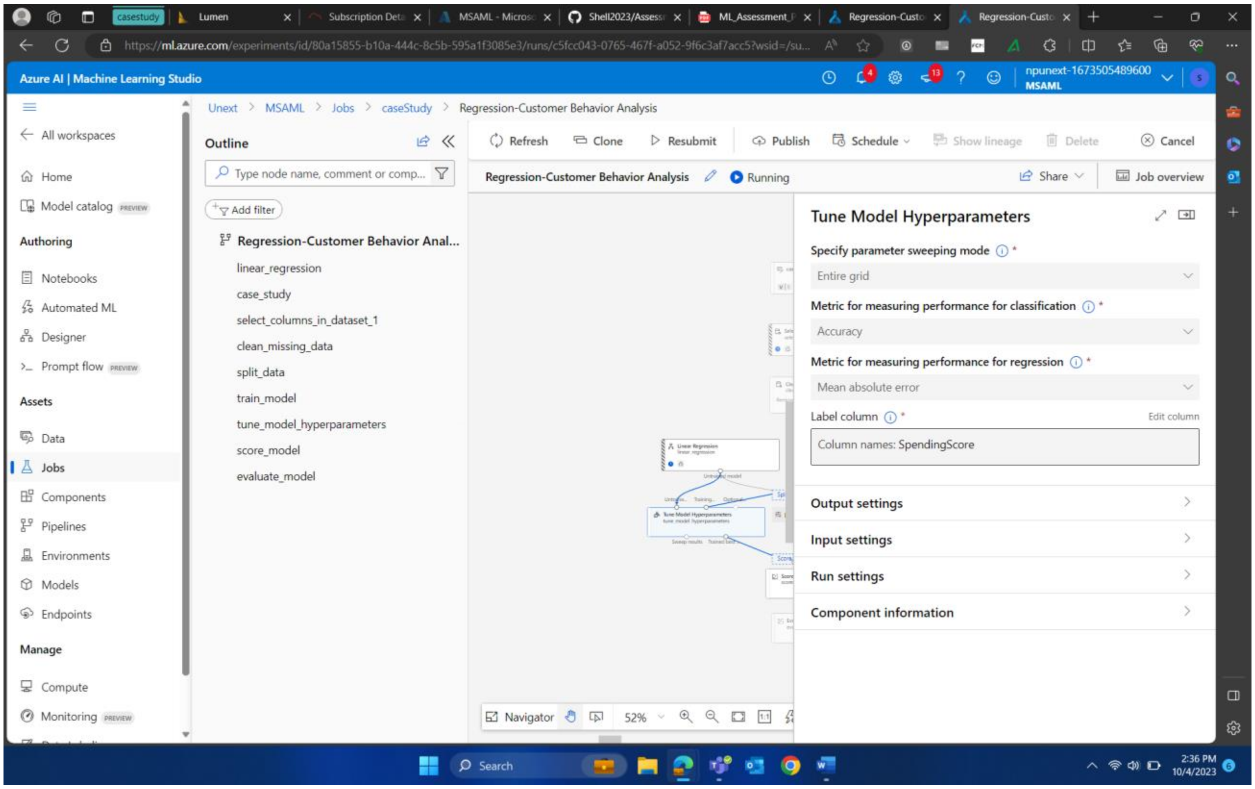
Outline

- Regression-Customer Behavior Anal...
  - linear_regression
  - case_study
  - select_columns_in_dataset_1
  - clean_missing_data
  - split_data
  - train_model
  - score_model
  - evaluate_model

3 a.



Hyper parameter tuning


Azure AI | Machine Learning Studio
Unext > MSAML > Jobs > caseStudy > Regression-Customer Behavior Analysis
Outline
Regression-Customer Behavior Anal...
linear_regression
case_study
select_columns_in_dataset_1
clean_missing_data
split_data
train_model
tune_model_hyperparameters
score_model
evaluate_model
Refresh
Clone
Resubmit
Publish
Schedule
Show lineage
Delete
Cancel
Regression-Customer Behavior Analysis
Running
Share
Job overview
Navigator
52%
Search
2:36 PM
10/4/2023

Tune Model Hyperparameters

Specify parameter sweeping mode *: Entire grid

Metric for measuring performance for classification *: Accuracy

Metric for measuring performance for regression *: Mean absolute error

Label column *: Column names: SpendingScore

Output settings

Input settings

Run settings

Component information

Assessment Questions.

1.
    a. Data Collection: Gather and acquire the data needed for training from various sources, including databases, files, web services, or external datasets.
    b. Data Cleaning: Identify and address issues like missing values, outliers, and inconsistencies within the dataset. This often involves actions such as filling in missing data, eliminating outlier values, or rectifying data errors.
    c. Data Transformation: Modify the data by actions like converting categorical variables into numerical forms, scaling numerical attributes, and encoding text or images.
    d. Feature Engineering: Enhance the dataset by creating new features or altering existing ones to improve model performance. This includes selecting relevant features and transforming them meaningfully.
    e. Data Splitting: Divide the dataset into distinct portions—typically training, validation, and test sets. The training set is used to train the model, the validation set aids in hyperparameter tuning, and the test set assesses model performance.
    f. Data Uploading: Upload the cleaned and processed dataset to Azure Machine Learning Workspace or storage, making it accessible for model training and experimentation.
    g. It is important to split dataset into training and validation because The training set is used to train the model, the validation set aids in hyperparameter tuning, and the test set assesses model performance.

2. Splitting a dataset into training and testing sets is essential in machine learning for various reasons:
    a. Model Assessment: It allows evaluating how well a model performs on new, unseen data.
    b. Overfitting Detection: Helps identify if a model is too closely tailored to the training data, indicating overfitting.
    c. Hyperparameter Tuning: Enables systematic tuning of model settings to optimize generalization.
    d. Performance Metrics: Provides data for calculating performance metrics like accuracy, precision, and recall.
    e. Bias and Variance Analysis: Aids in assessing bias and variance to strike a balance.
    f. Model Selection: Facilitates choosing the best model among candidates.
    g. Validation of Assumptions: Ensures model behavior aligns with expectations and real-world scenarios.
    h. Real-world Simulation: Mimics the model's performance in production environments.

3. We can see that there is a linear trend or direct relation between the income and spending score and age, we can try using linear regression model.
    a. Interpretability: Linear Regression provides clear interpretability of coefficients. This means we can easily understand the direction and magnitude of the impact of each predictor variable on the purchasing behavior. This is important when we want to explain why certain factors influence buying decisions.

   b. Simplicity: Linear Regression is a simple and straightforward algorithm. It's easy to implement and understand, making it a good choice when we want to quickly analyze the relationship between a few key variables and purchasing behavior.

   Linear Regression is a suitable algorithm for predicting customer purchasing behavior when we have a reasonable expectation of a linear relationship between predictor variables and the target variable. It offers interpretability and simplicity, making it a valuable tool in certain scenarios. However, it's essential to assess whether its assumptions hold and whether it provides adequate predictive performance for your specific dataset and goals.

4. Hyperparameter tuning involves optimizing a machine learning model's settings, called hyperparameters, for better performance. It's vital because:
   a. Improved Performance: Proper hyperparameter choices enhance model accuracy and prevent overfitting.
   b. Resource Efficiency: Efficient hyperparameters save training time and resources.
   c. Robustness: Tuned hyperparameters make models adaptable to diverse data

   Grid Search is a systematic technique:

   a. It explores hyperparameter combinations, such as learning rates and tree numbers, exhaustively.
   b. Cross-validation assesses each combination's performance, finding the best setup.
   c. Benefits include thorough exploration, reproducibility, and ease of implementation. However, it can be computationally intensive.