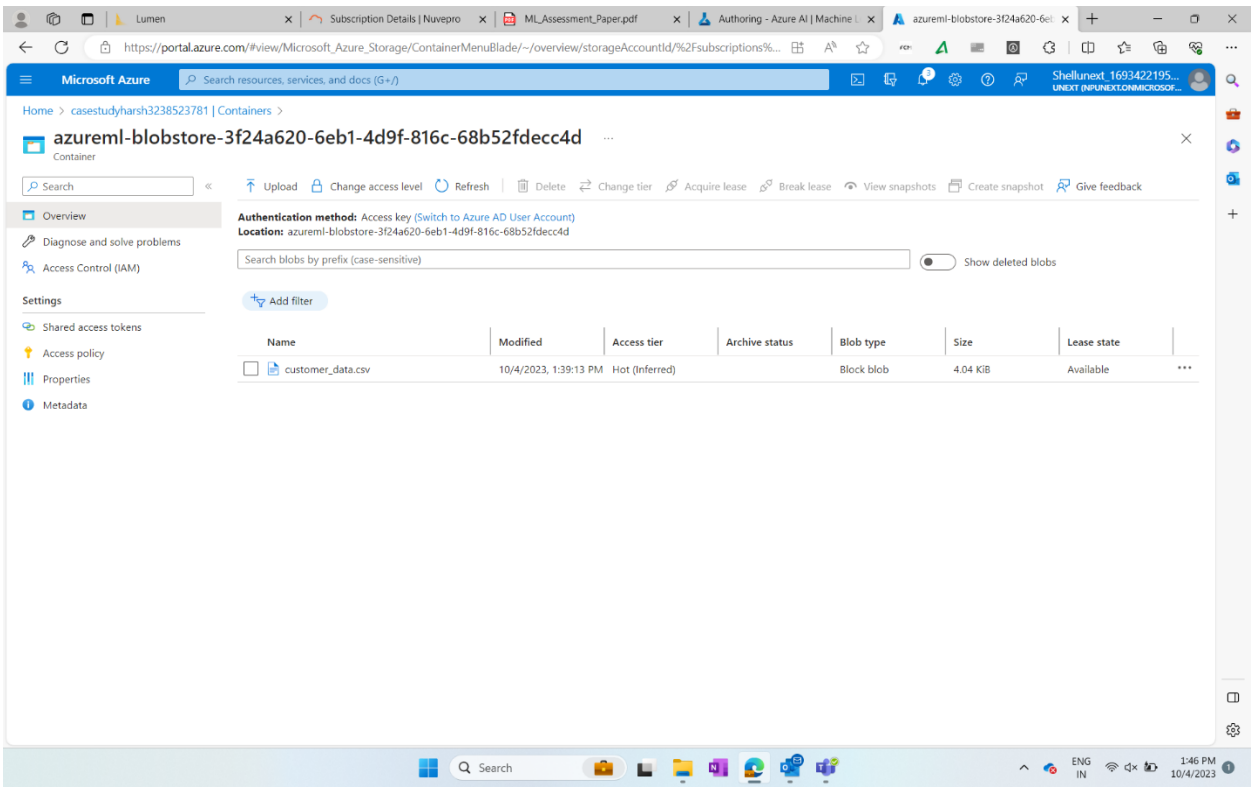
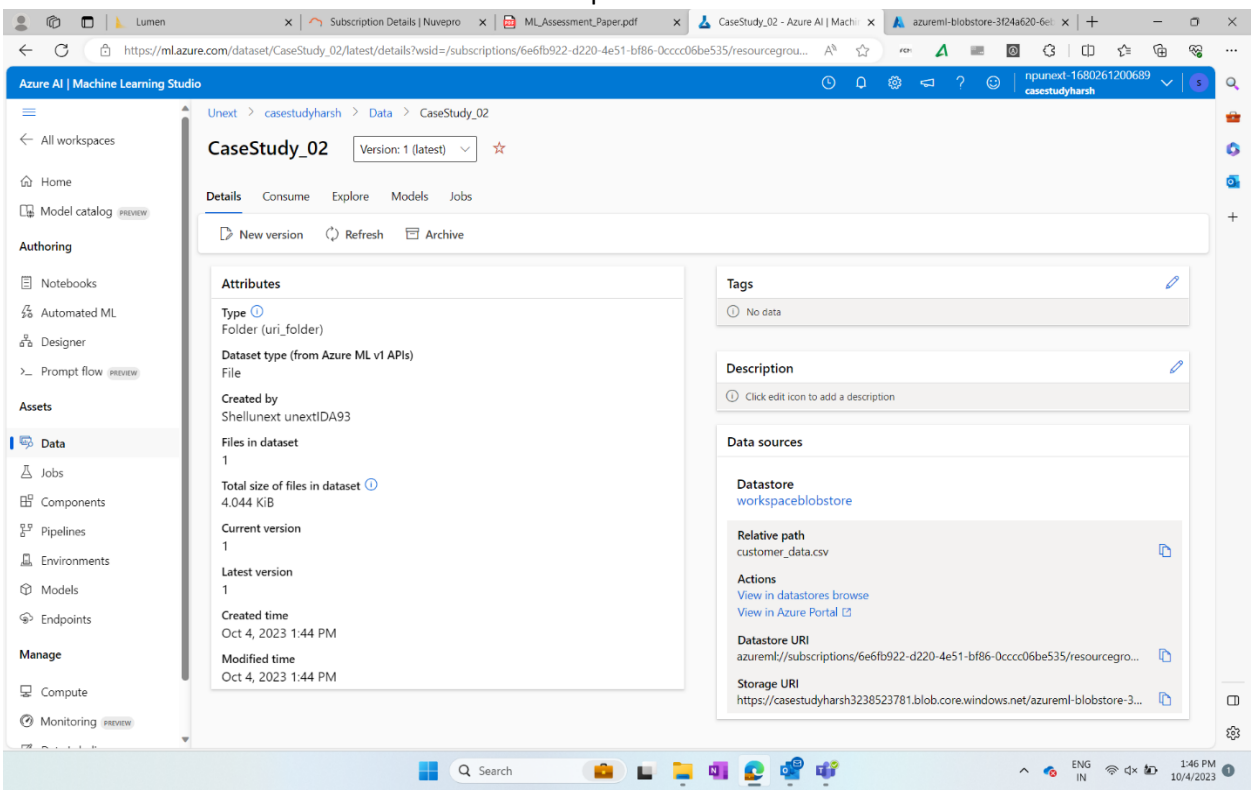
Case Study (ML)

Screenshots of Azure Machine Learning Studio

AzureML Blob Storage

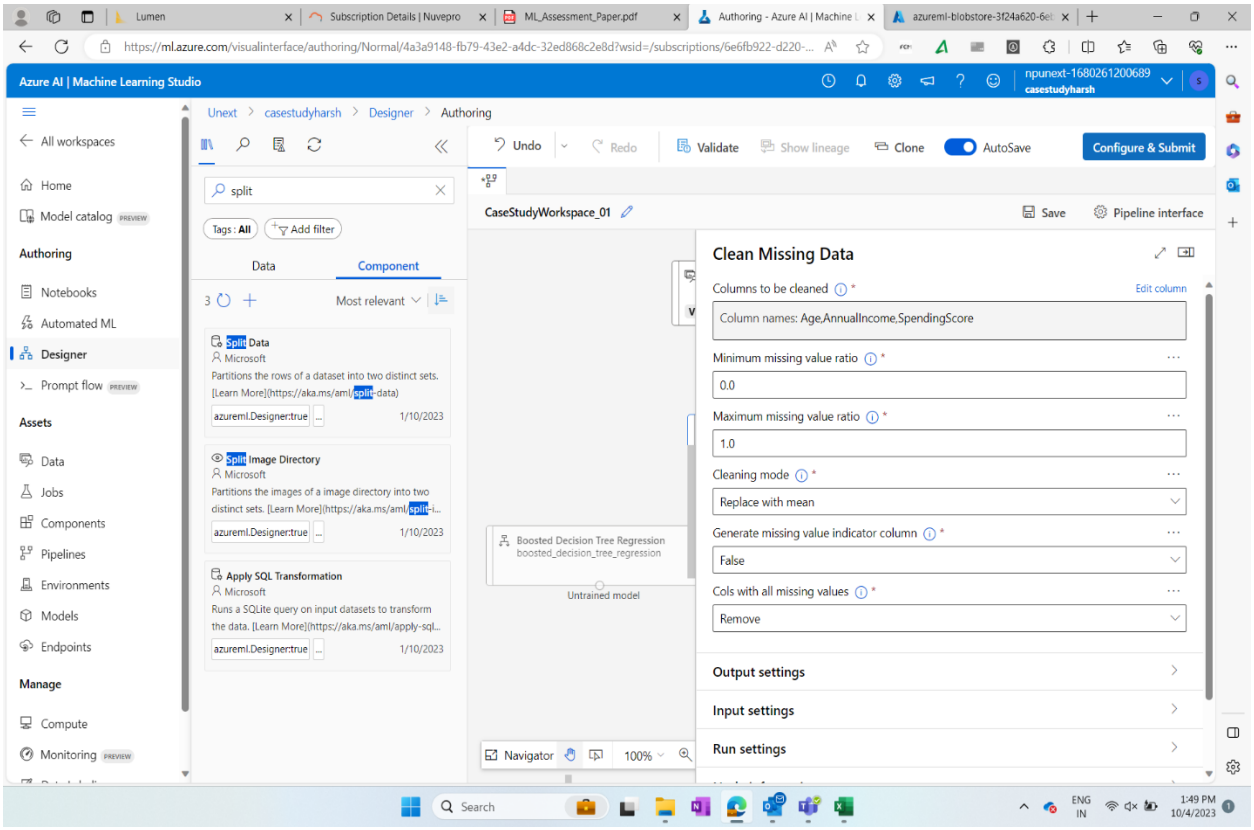


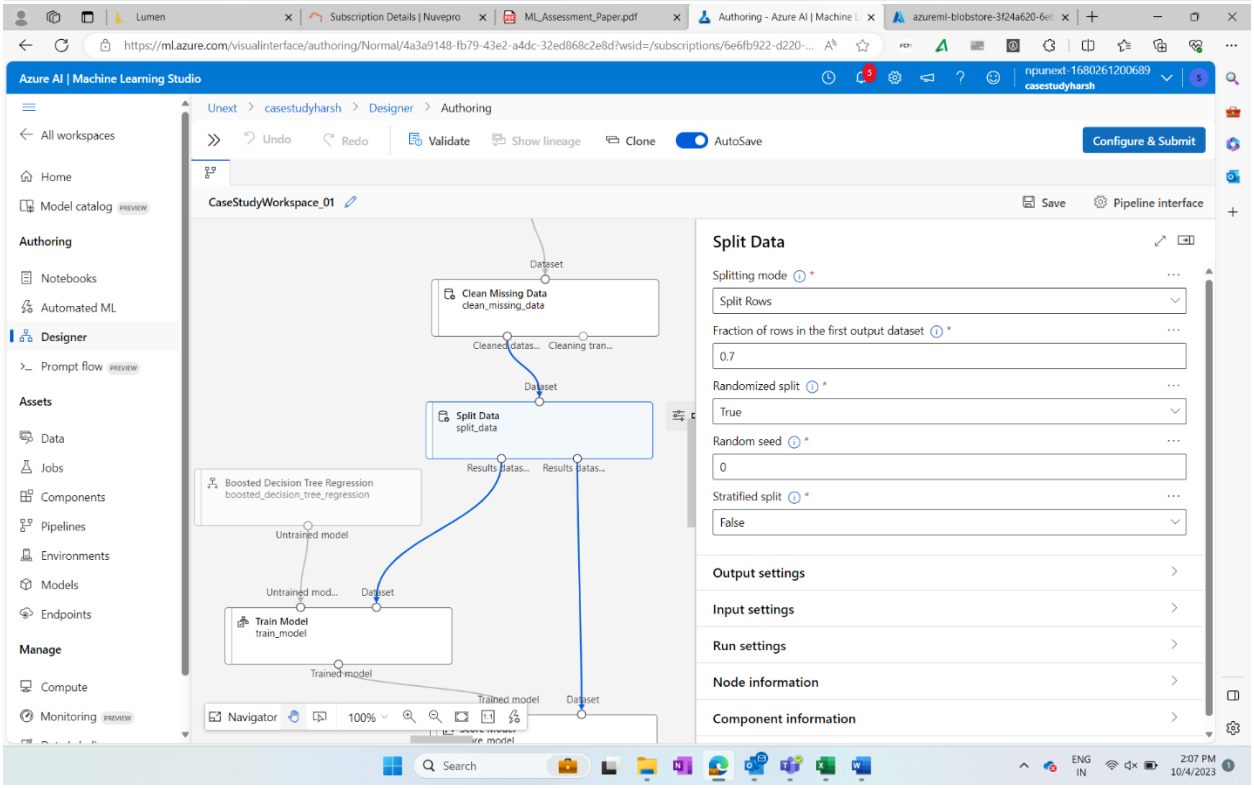
Creation of data asset on AzureML Workspace

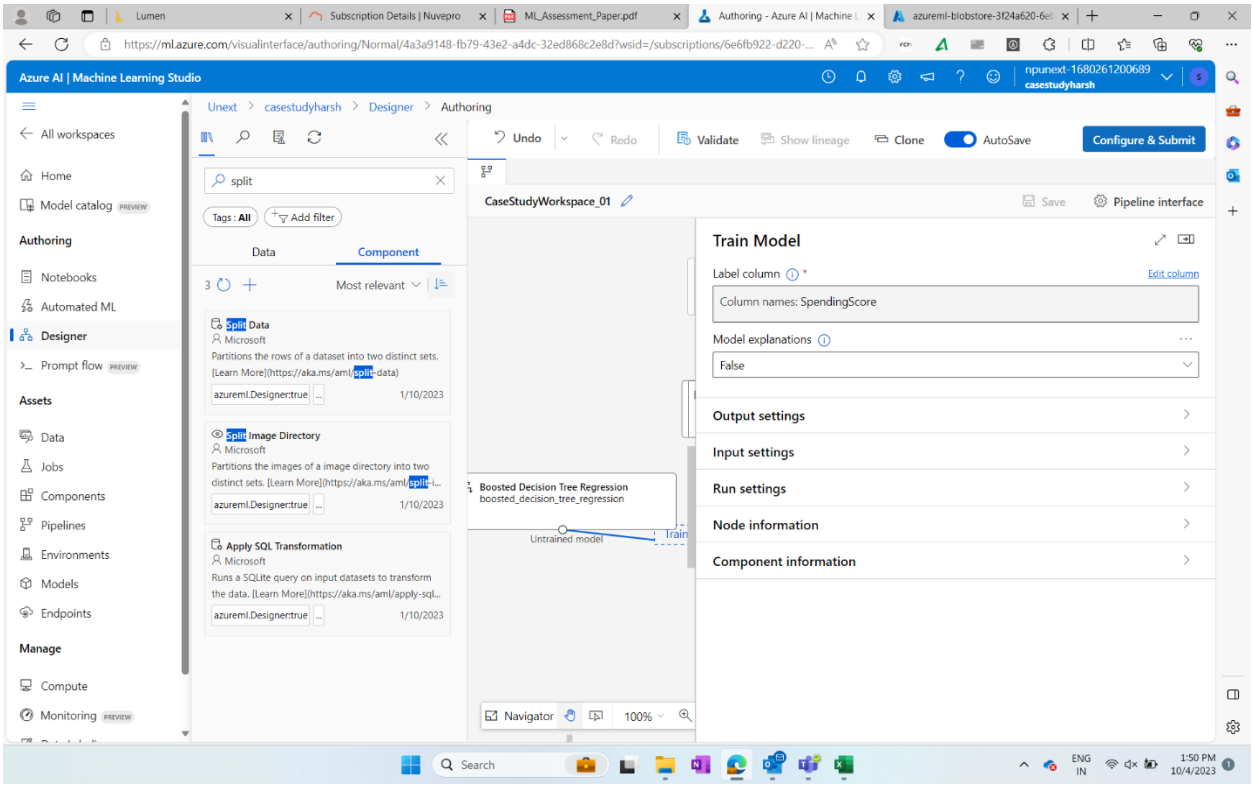


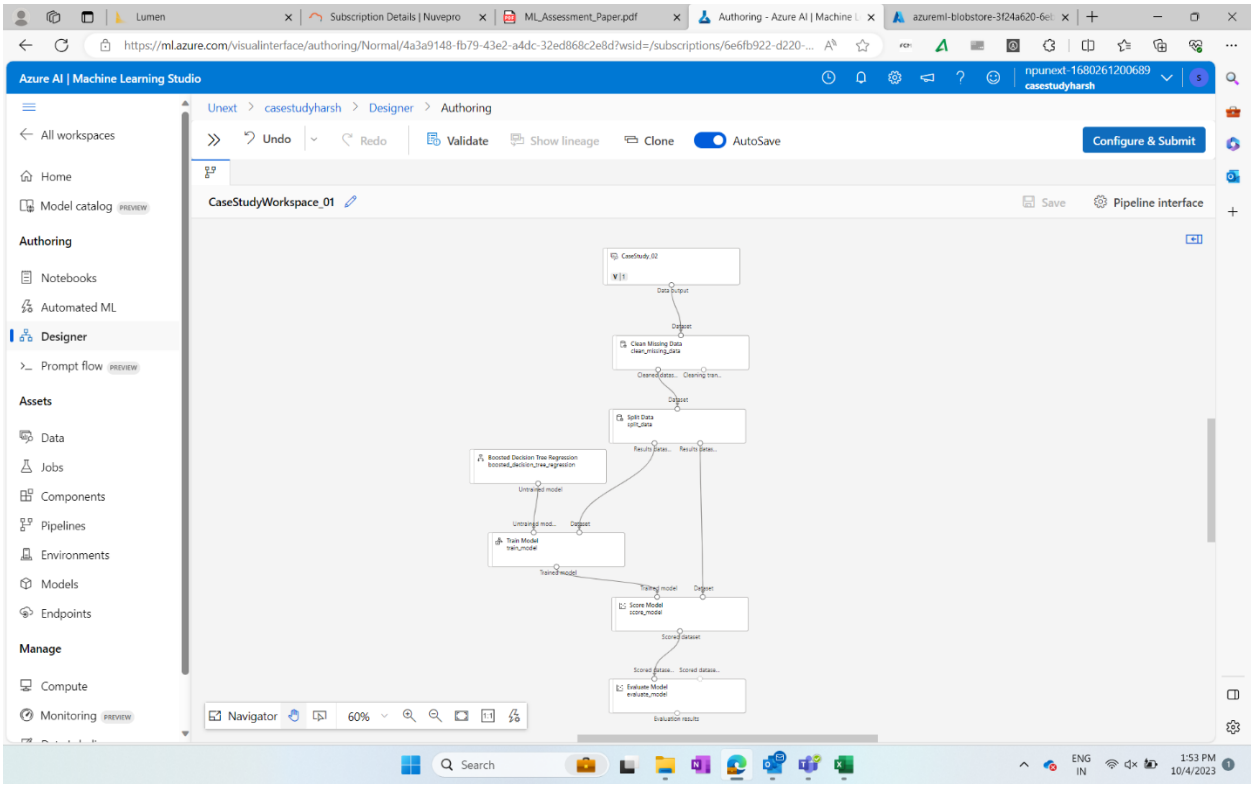
Configuring the workspace:

Train model:

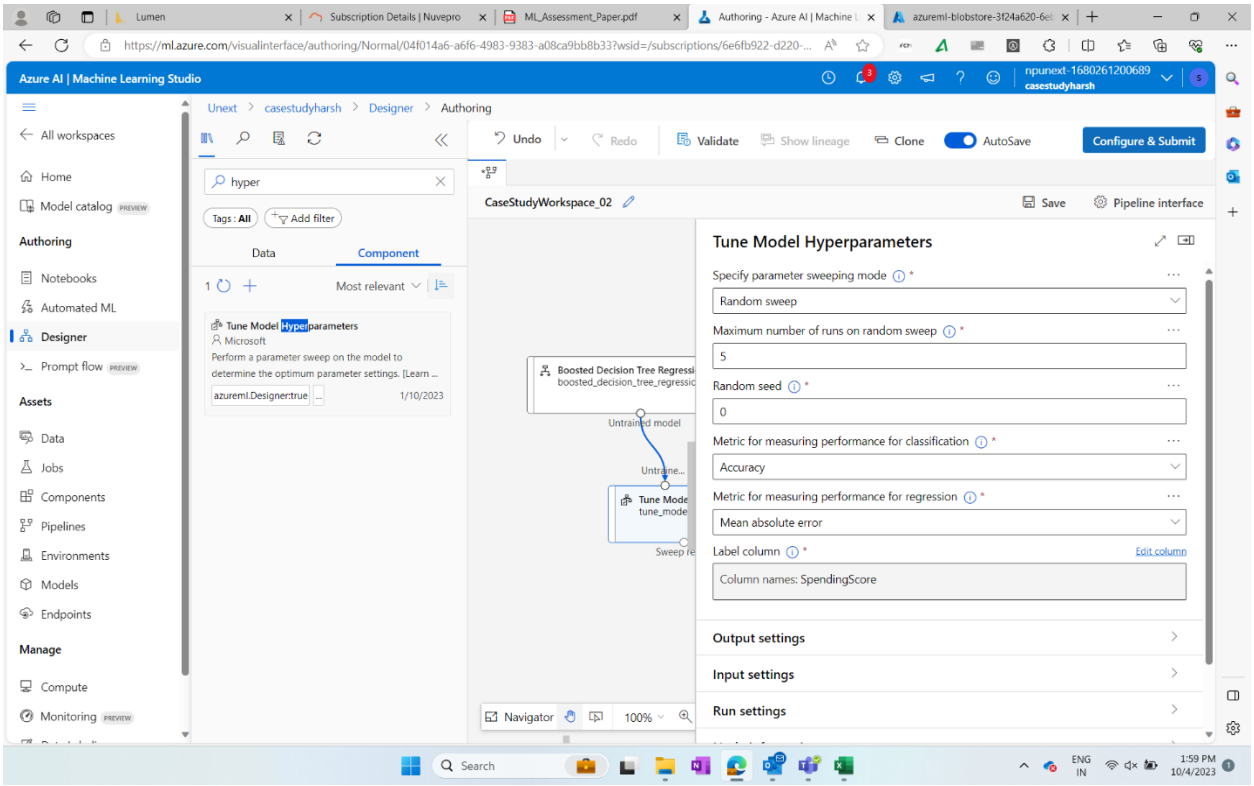


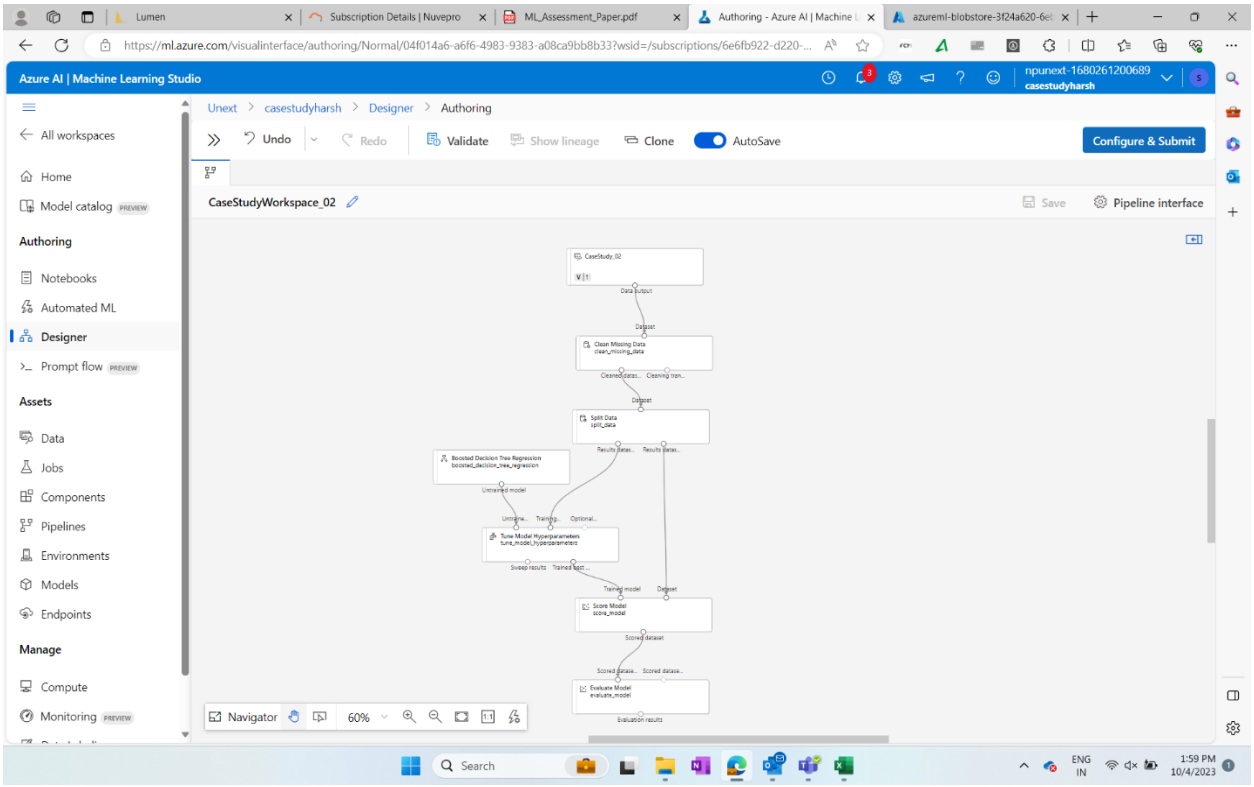



Azure AI | Machine Learning Studio
Unext > casestudyharsh > Designer > Authoring
All workspaces
Home
Model catalog
Authoring
Notebooks
Automated ML
Designer
Prompt flow
Assets
Data
Jobs
Components
Pipelines
Environments
Models
Endpoints
Manage
Compute
Monitoring
Undo
Redo
Validate
Show lineage
Clone
AutoSave
Configure & Submit
CaseStudyWorkspace_01
Save
Pipeline interface
split
Tags : All
Add filter
Data
Component
Most relevant
Split Data
Microsoft
Partitions the rows of a dataset into two distinct sets.
Split Image Directory
Partitions the images of a image directory into two distinct sets.
Apply SQL Transformation
Runs a SQLite query on input datasets to transform the data.
Boosted Decision Tree Regression
boosted_decision_tree_regression
Untrained model
Train Model
Label column
Edit column
Column names: SpendingScore
Model explanations
False
Output settings
Input settings
Run settings
Node information
Component information
Navigator
100%


Azure AI | Machine Learning Studio
Unext > casestudyharsh > Designer > Authoring
All workspaces
Home
Model catalog
Authoring
Notebooks
Automated ML
Designer
Prompt flow
Assets
Data
Jobs
Components
Pipelines
Environments
Models
Endpoints
Manage
Compute
Monitoring
Undo
Redo
Validate
Show lineage
Clone
AutoSave
Configure & Submit
CaseStudyWorkspace_01
Save
Pipeline interface
Navigator
60%

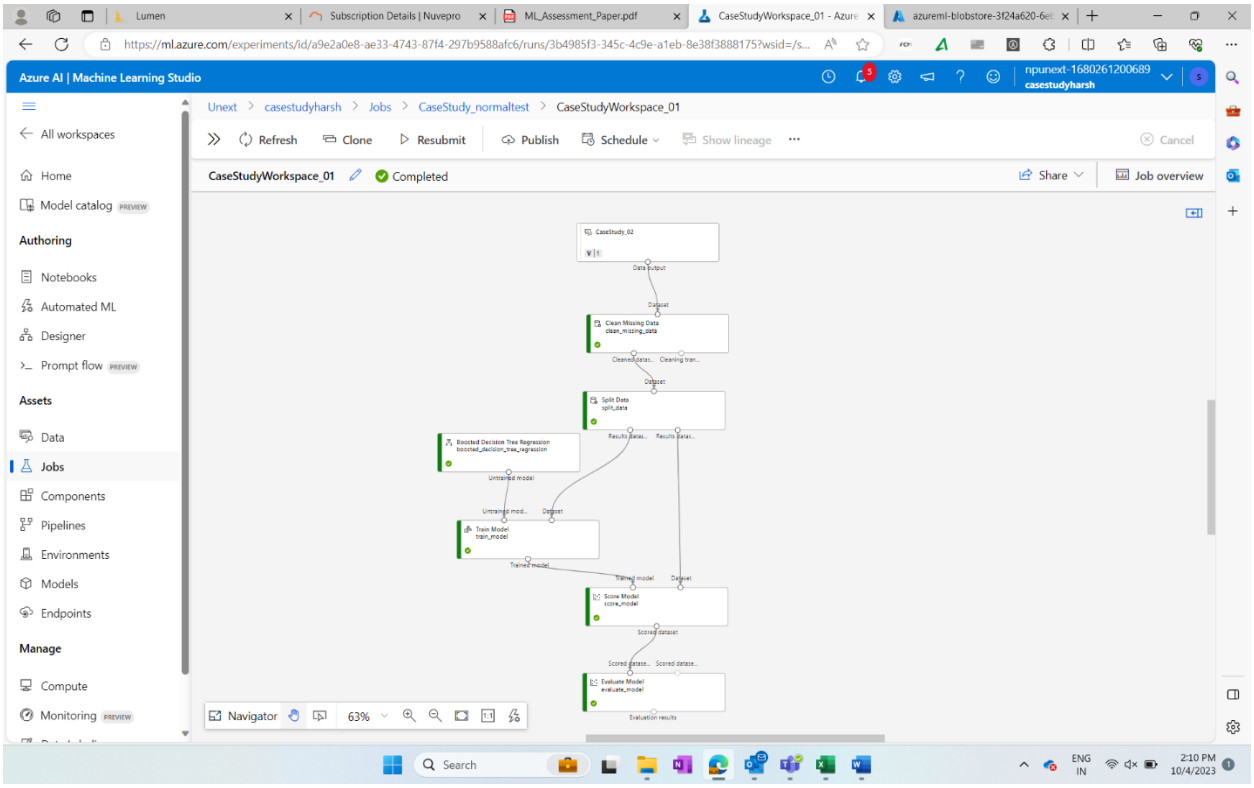
Hyperparameter training model:

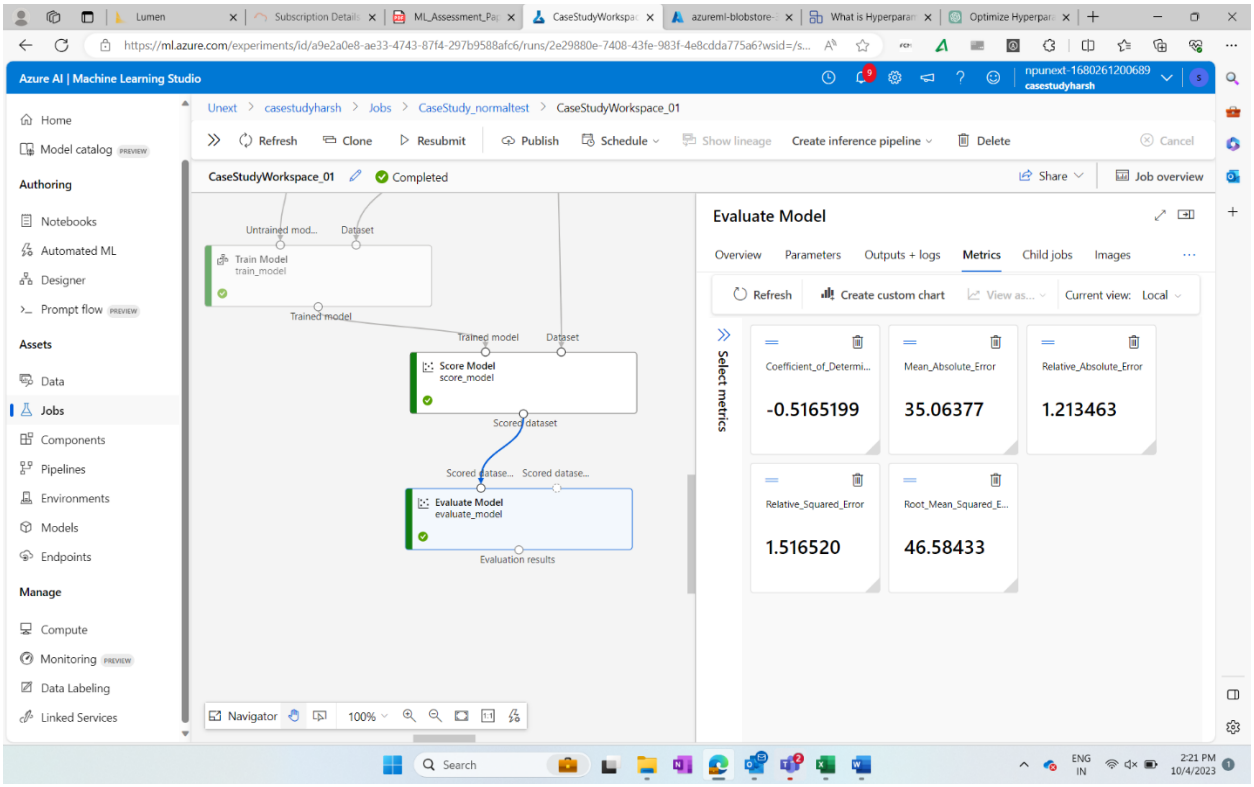



Lumen
Subscription Details | Nuvepro
ML_Assessment_Paper.pdf
Authoring - Azure AI | Machine L
azureml-blobstore-3f24a620-6eb
https://ml.azure.com/visualinterface/authoring/Normal/04f014a6-a6f6-4983-9383-a08ca9bb8b33?wsid=/subscriptions/6e6fb922-d220-...
Azure AI | Machine Learning Studio
npunext-1680261200689
casestudyharsh
All workspaces
Home
Model catalog PREVIEW
Authoring
Notebooks
Automated ML
Designer
Prompt flow PREVIEW
Assets
Data
Jobs
Components
Pipelines
Environments
Models
Endpoints
Manage
Compute
Monitoring PREVIEW
Unext > casestudyharsh > Designer > Authoring
Undo
Redo
Validate
Show lineage
Clone
AutoSave
Configure & Submit
CaseStudyWorkspace_02
Save
Pipeline interface
CaseStudy_02
Data output
Dataset
Clean Missing Data
clean_missing_data
Cleaned datas...
Cleaning tran...
Dataset
Split Data
split_data
Results dates...
Results dates...
Boosted Decision Tree Regression
boosted_decision_tree_regression
Untrained model
Untrainе...
Training...
Optional...
Tune Model Hyperparameters
tune_model_hyperparameters
Sweep results
Trained best...
Trained model
Dataset
Score Model
score_model
Scored dataset
Scored datase...
Scored datase...
Evaluate Model
evaluate_model
Evaluation results
Navigator
60%
Search
ENG
IN
1:59 PM
10/4/2023

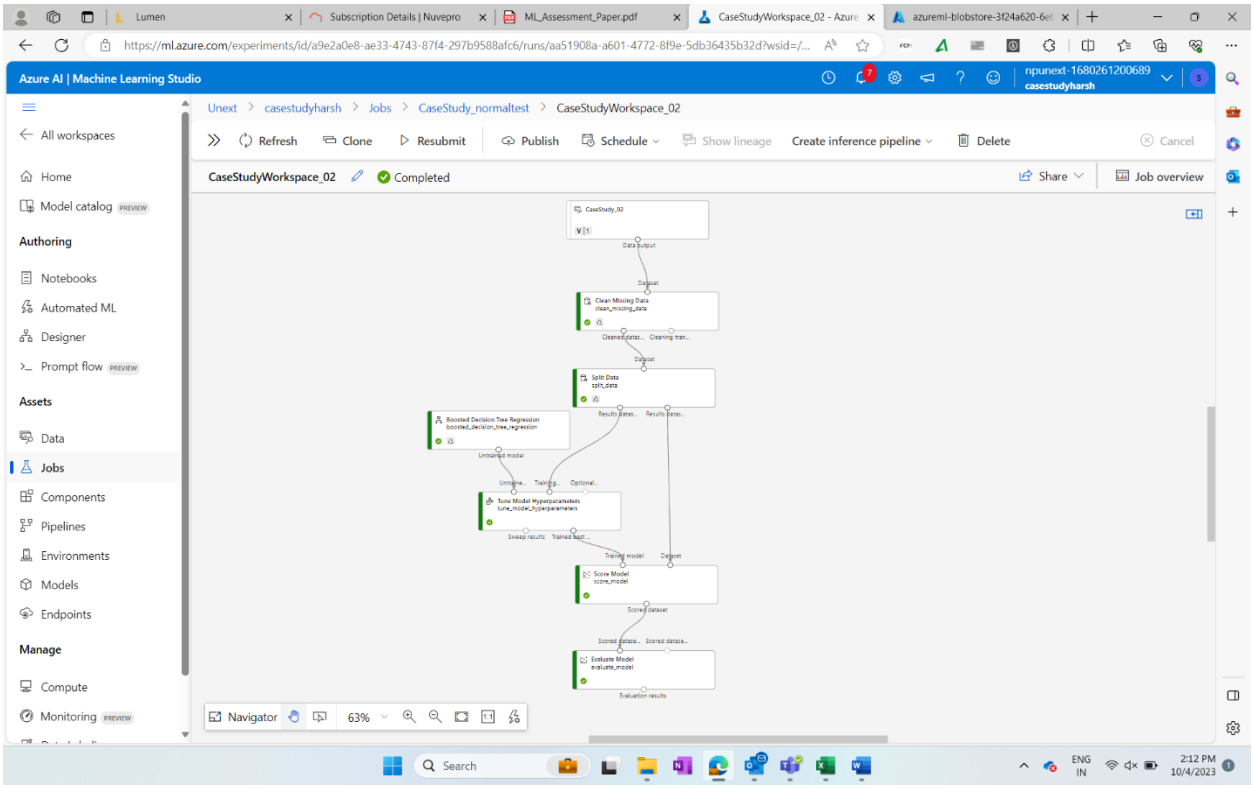
Outputs:

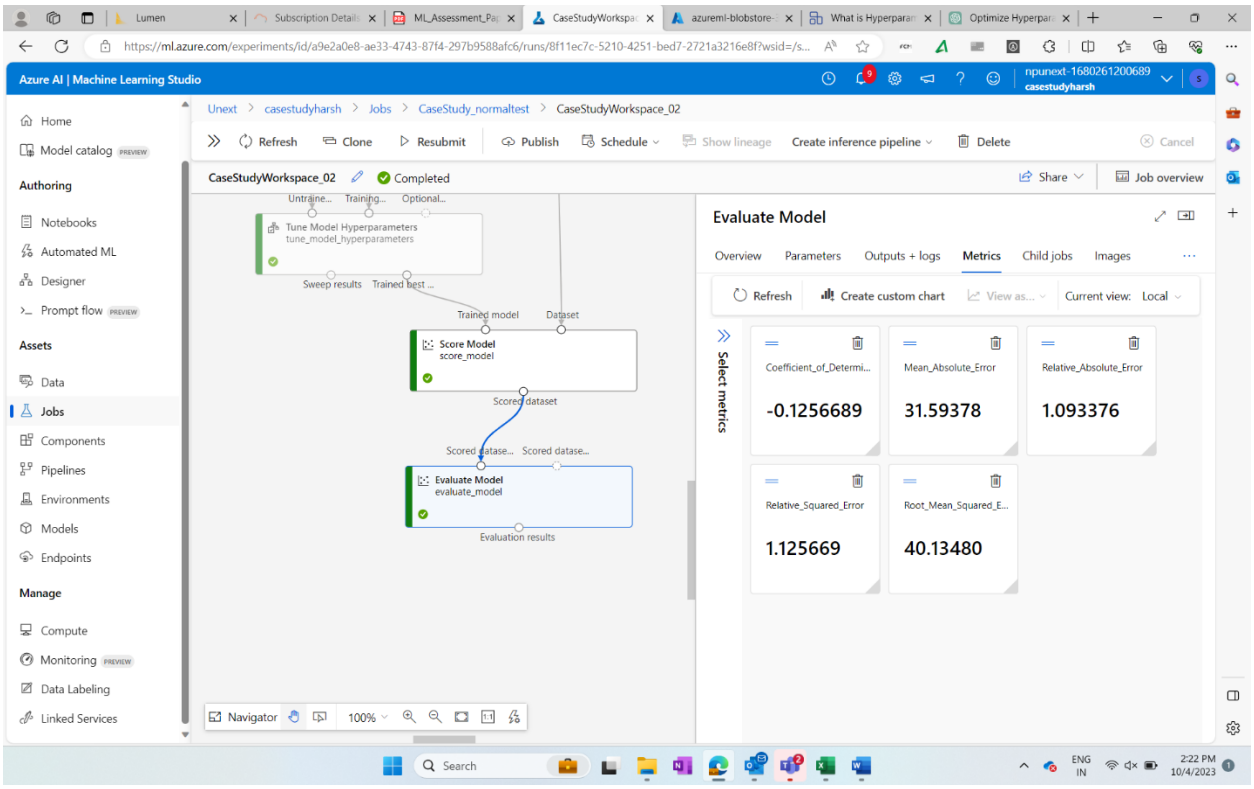
Train model:





Hyperparameter Training:





Question/Answers

1. Firstly, the data set is uploaded through blob storage account to pipeline. Then, Clean_Missing_Data is used to clear out any missing values, so that the garbage data is removed and only the data with relevant information is kept for training. Then, the data is split into training and testing data with 70% kept for training and 30% for testing. Finally, the split training data goes for training using the ML algorithm “Boosted Decision Tree Regression”.
2. It is important to split the dataset into training and testing sets as the model shouldn’t be tested with the same data that it has been trained with, as that won’t test its capabilities and it would be impossible to check whether the model has been trained to recognize patterns in new, unfamiliar data. This helps in model evaluation as the models performance is tested on separate data than what it’s used to.
3. I have chosen Boosted decision tree algorithm for this problem as there was only one csv file, which had large amount of data. We can akso use linear regression for the same, but I believe that a decision tree in this case would be better.
4. Hyperparameter tuning involves finding the best hyperparameter values for a machine learning algorithm to improve its performance. This optimization aims to minimize a predefined loss function, leading to better results with fewer errors.
   Random search, as opposed to grid search, randomly explores hyperparameter combinations and returns the best-performing one after multiple iterations. It's efficient for handling many hyperparameters and wide search ranges, offering faster results without user-defined biases. However, it may not guarantee the absolute best hyperparameter combination due to its random approach.