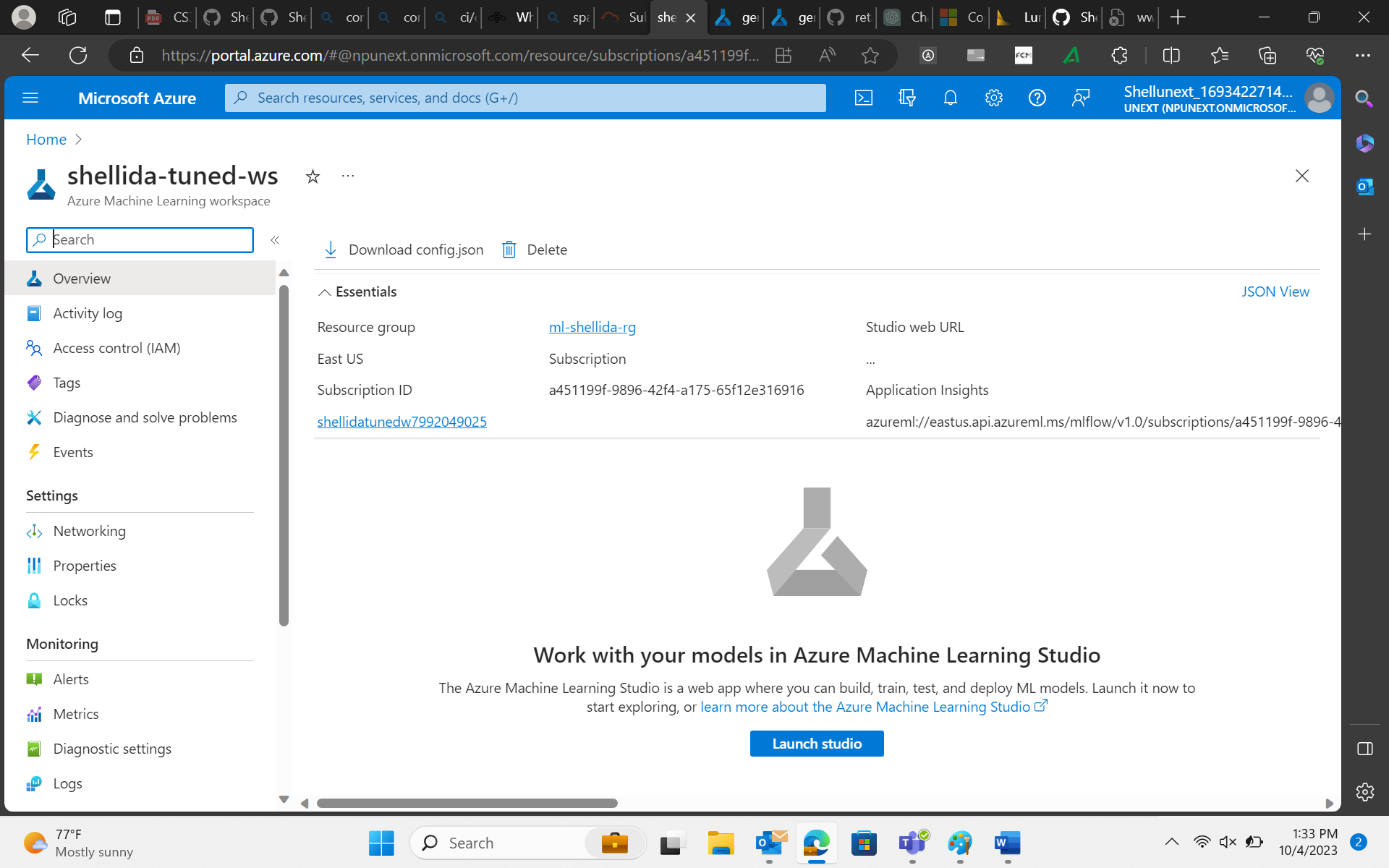
**Azure ML Studio Hands-on Assessment**

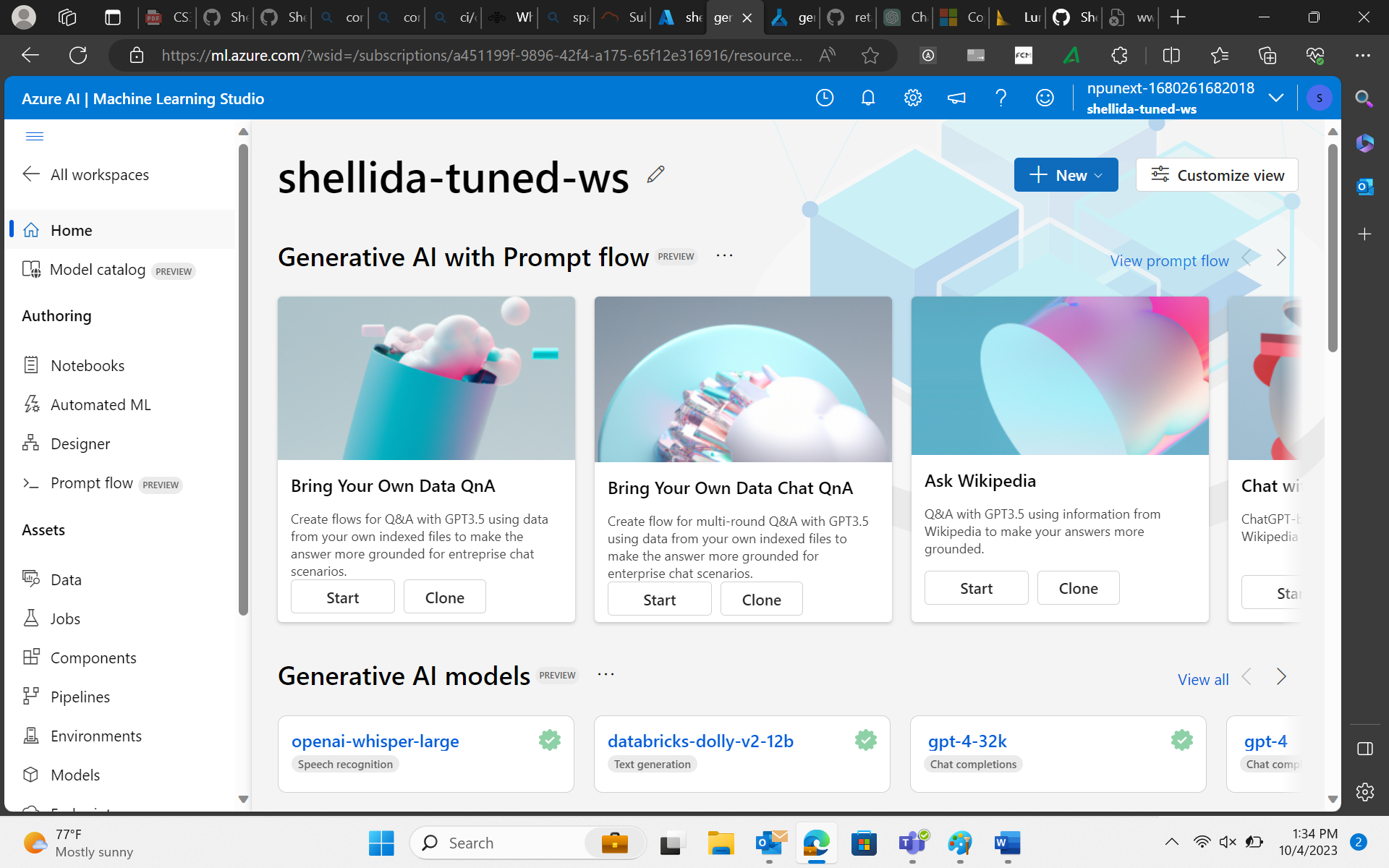
**Manya Agarwal**

**Emp Id : 654801**

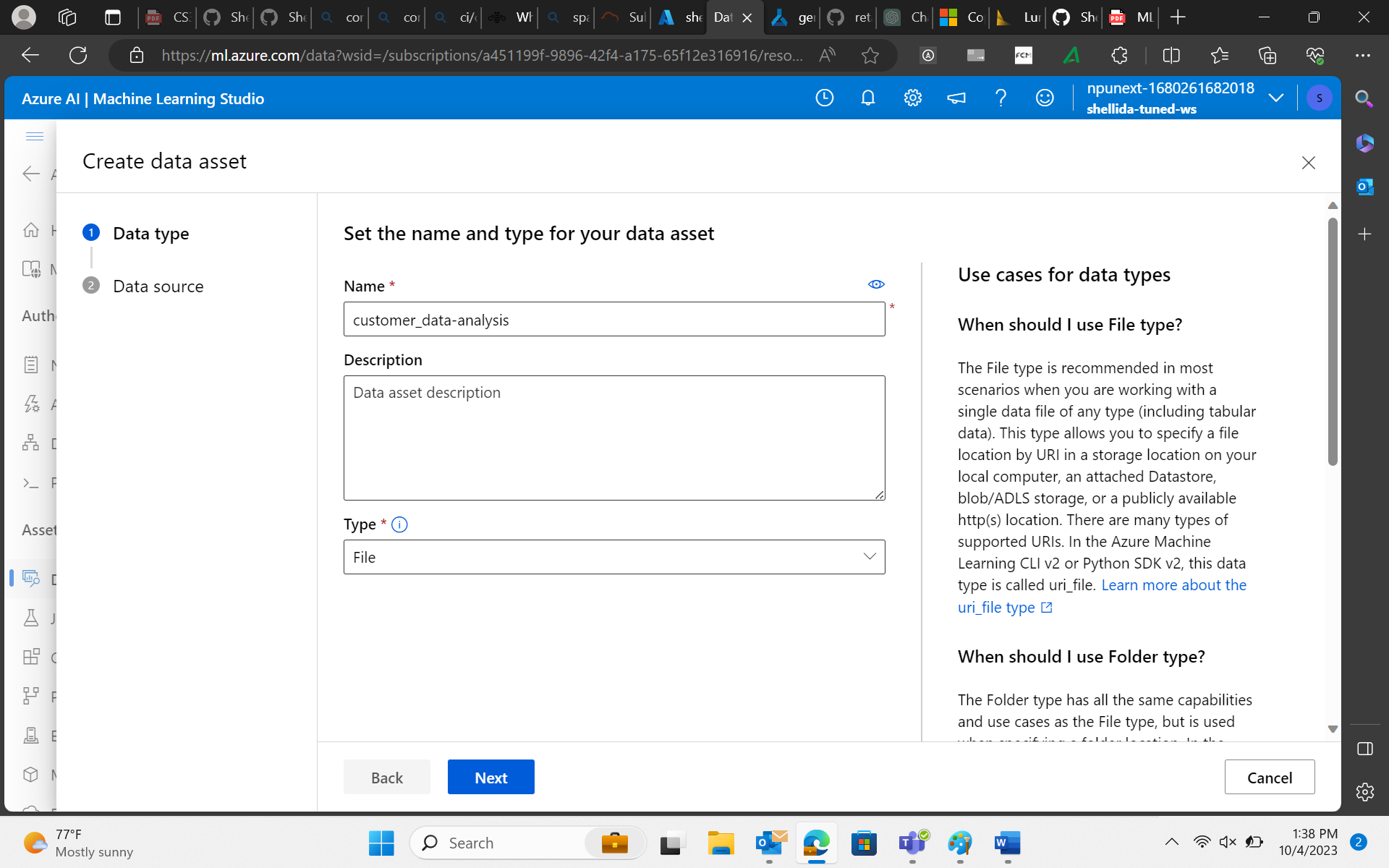
1. Launch Azure Machine Learning in Azure Portal.
2. Create a new workspace – shellida-tuned-ws.



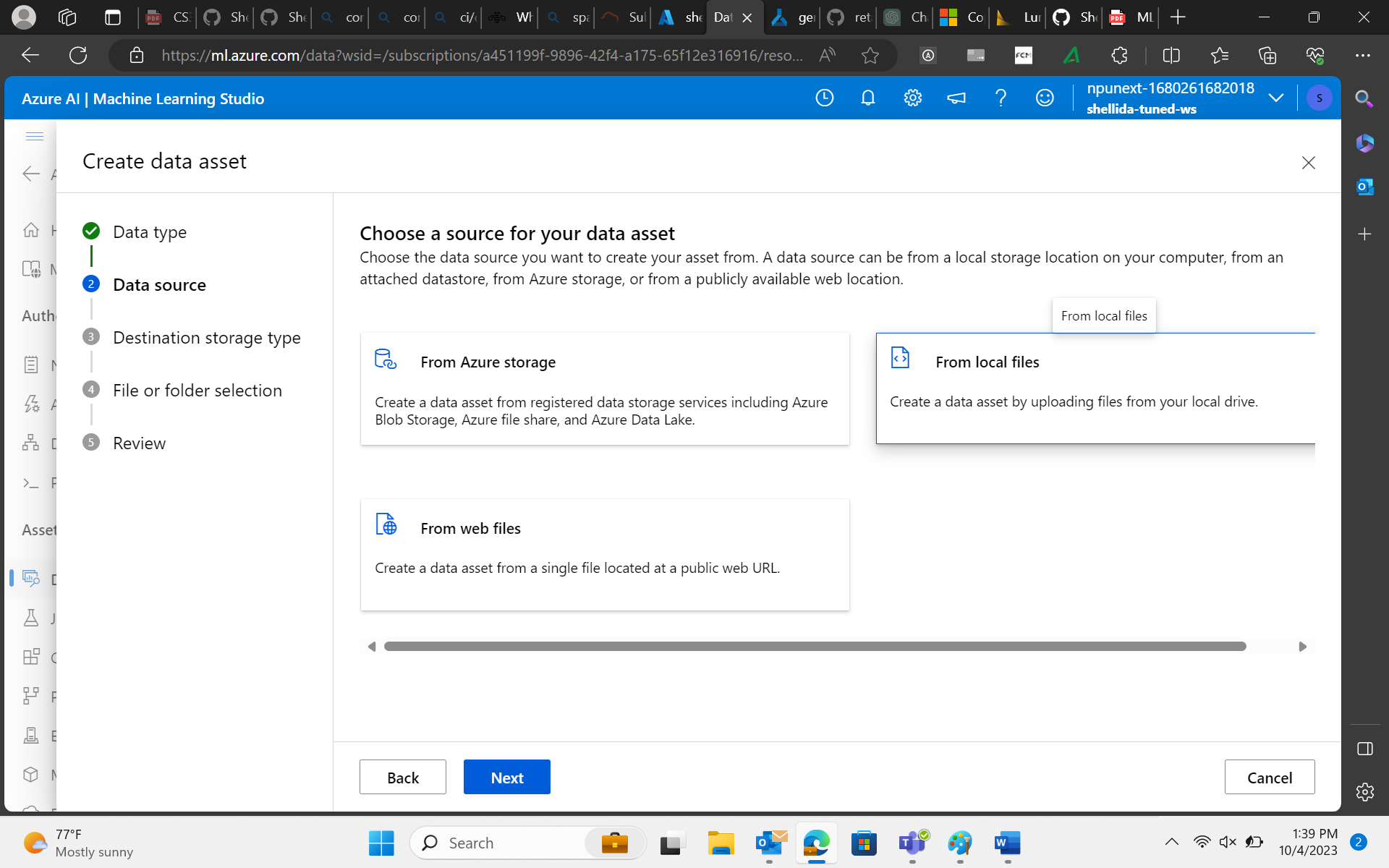
1. Launch Studio

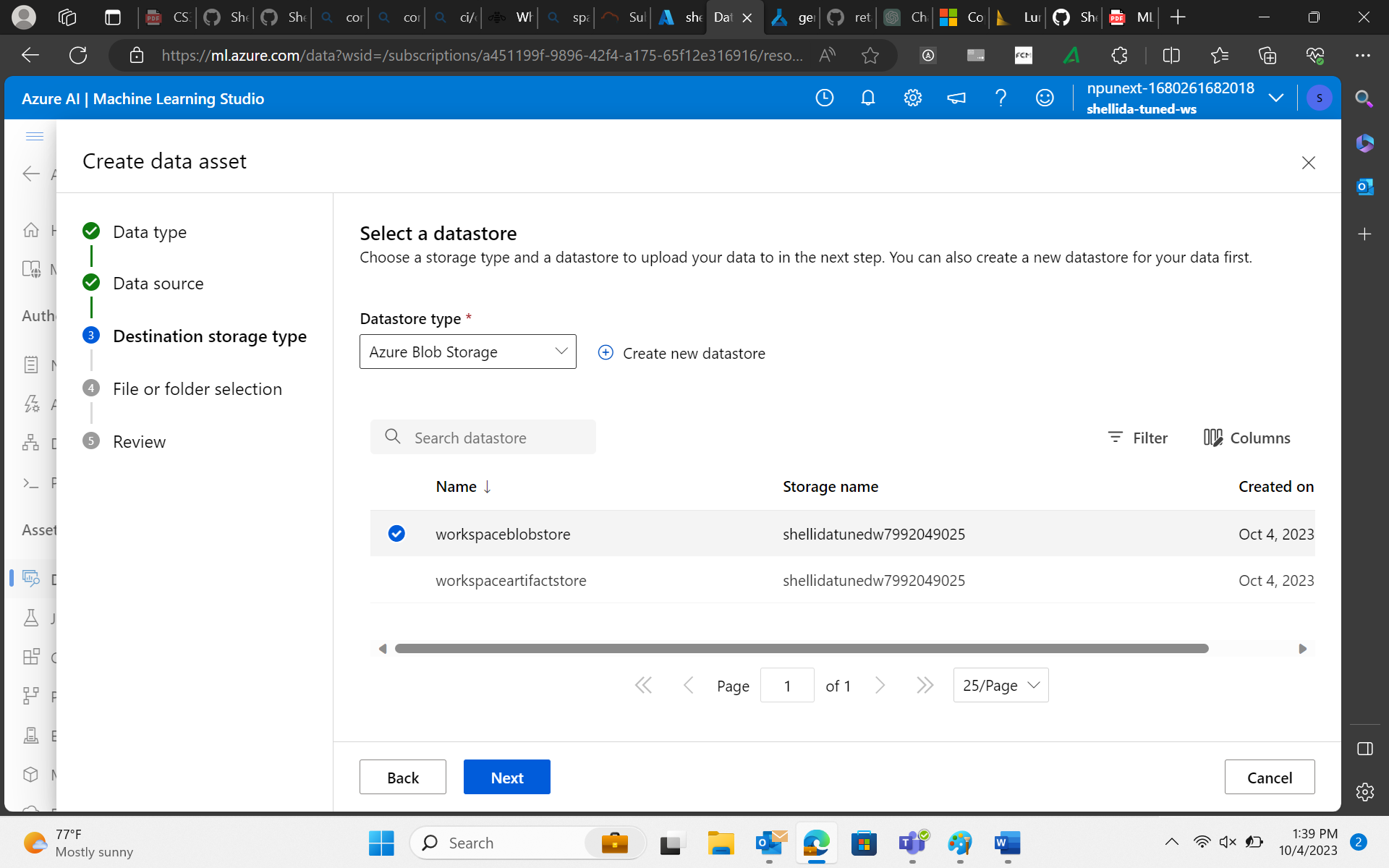


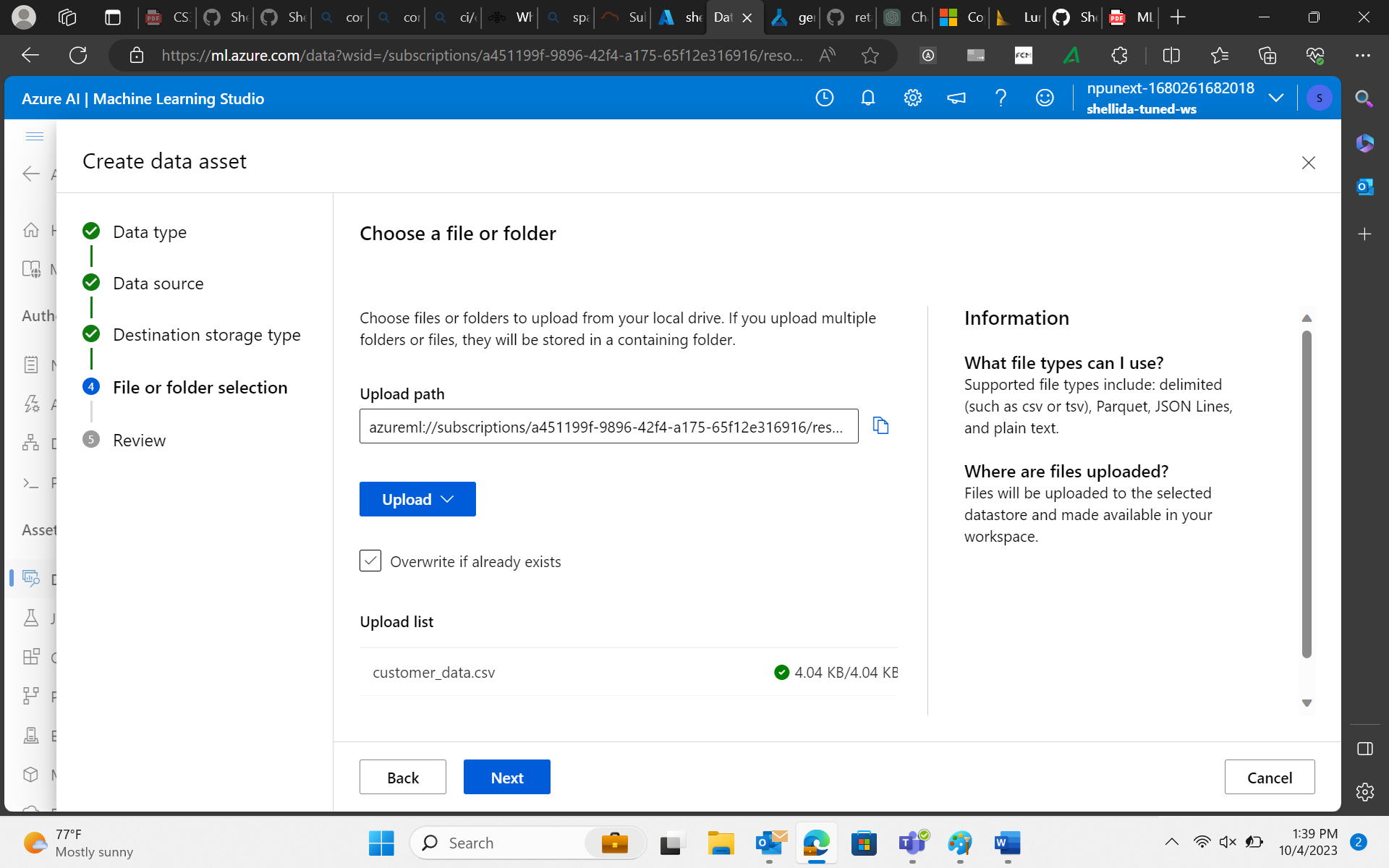
1. Create Data Asset customer\_data-analysis



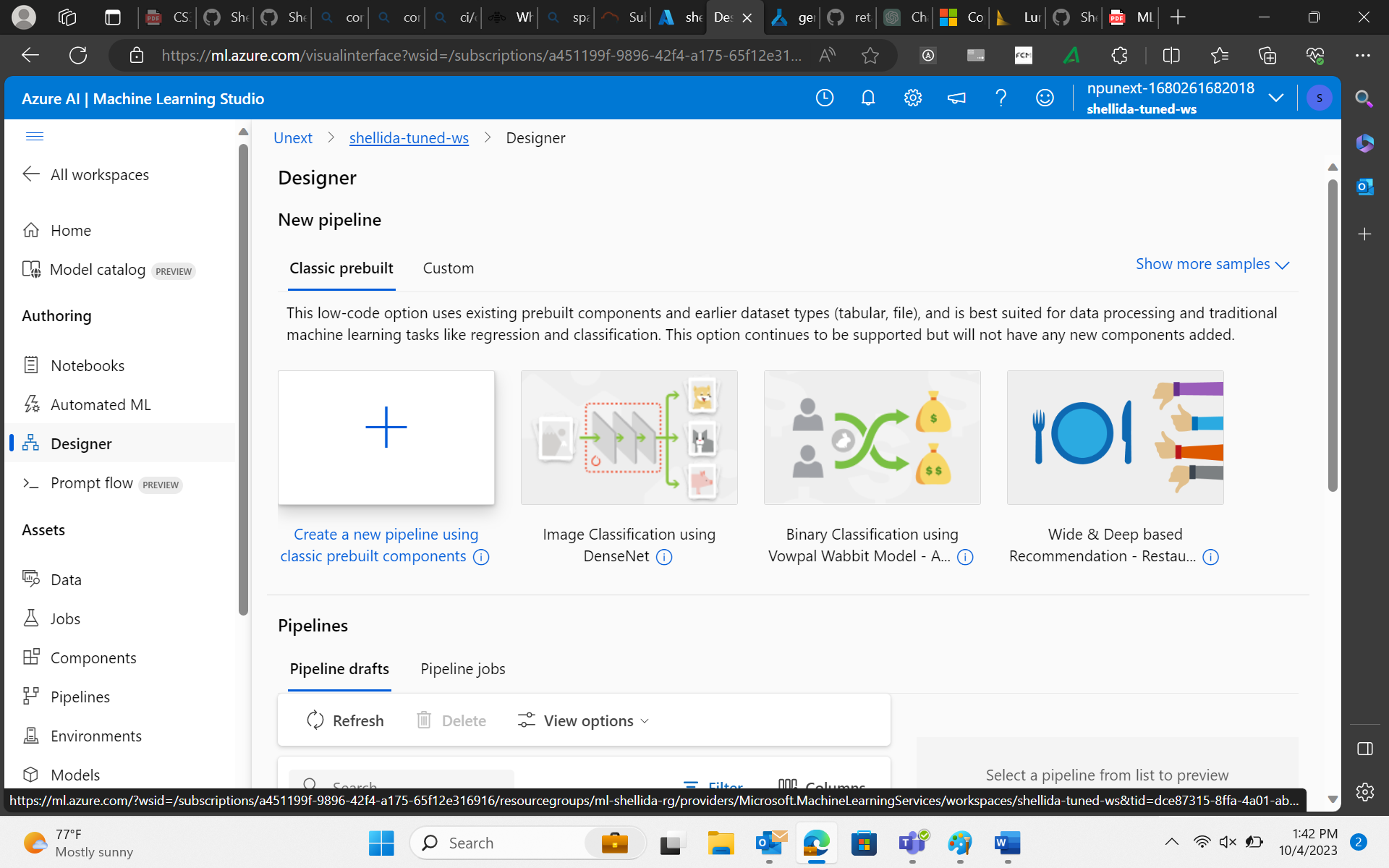
1. Choose data from local file.



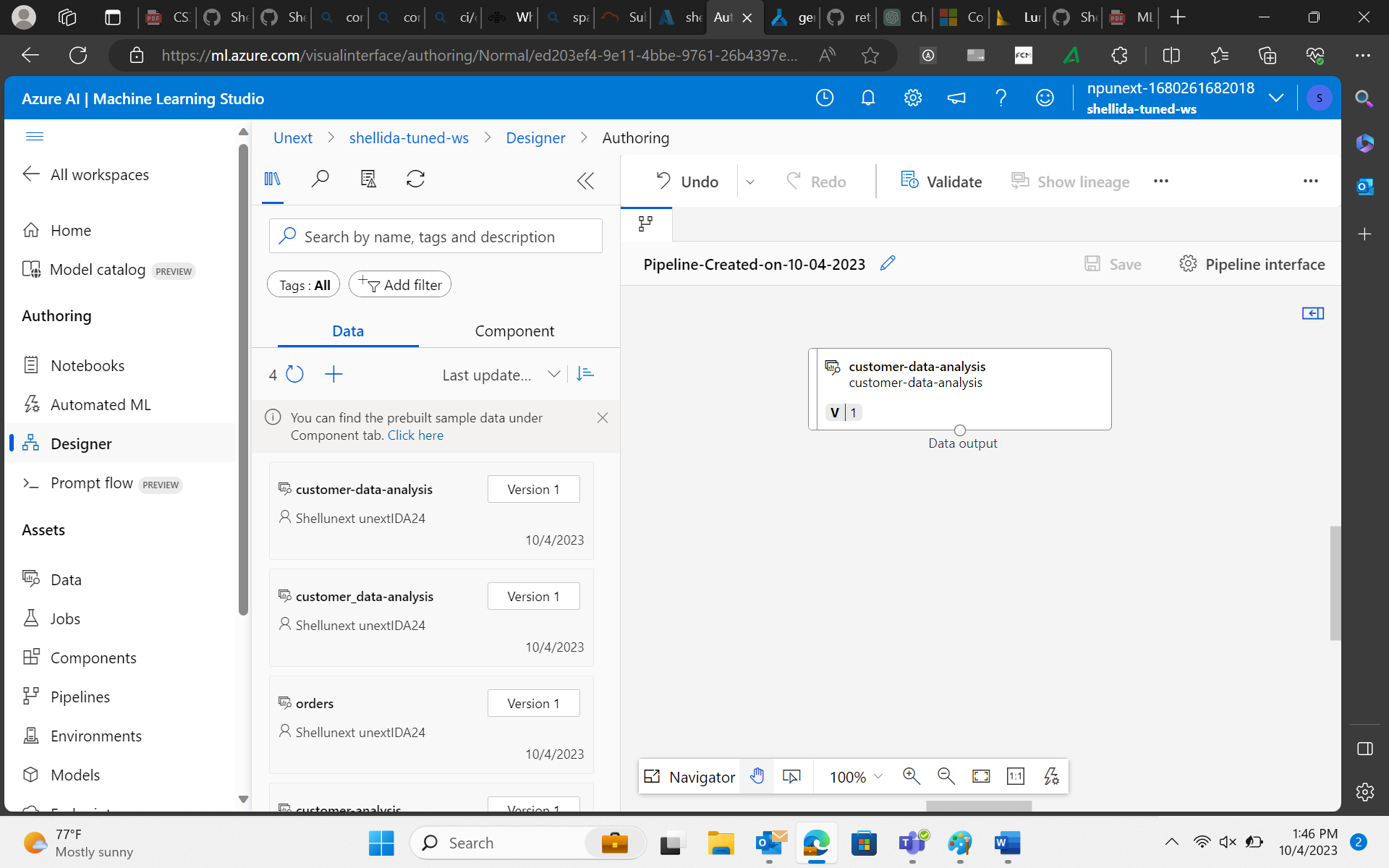




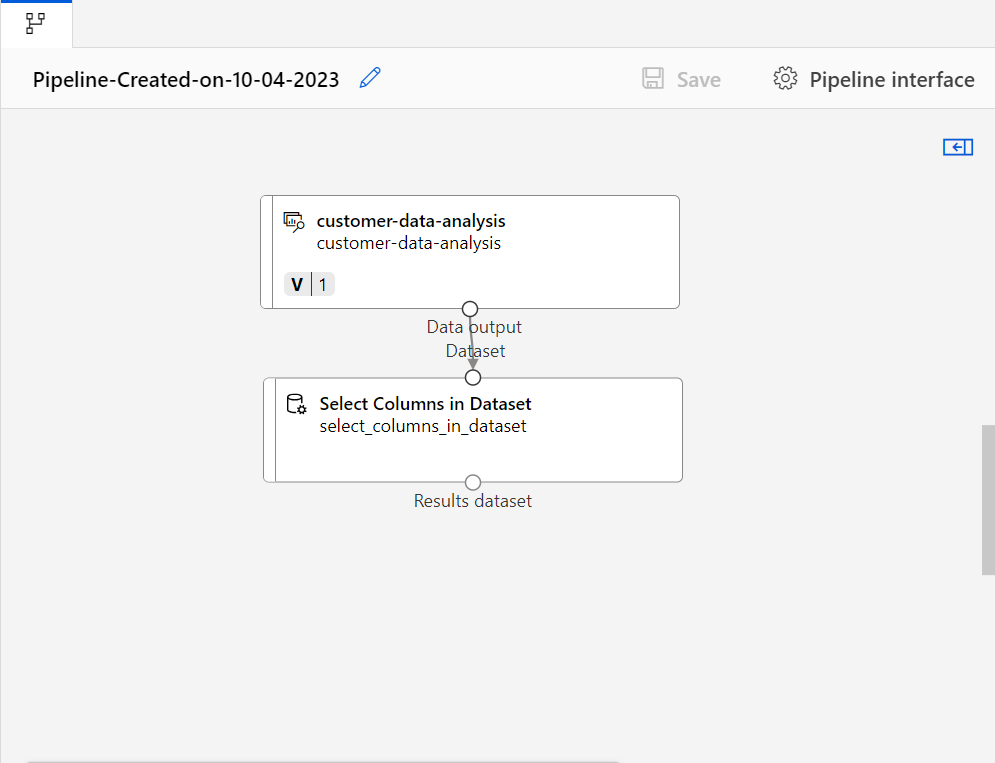
1. Open Designer in Azure Machine Learning.
2. Create a new pipeline.

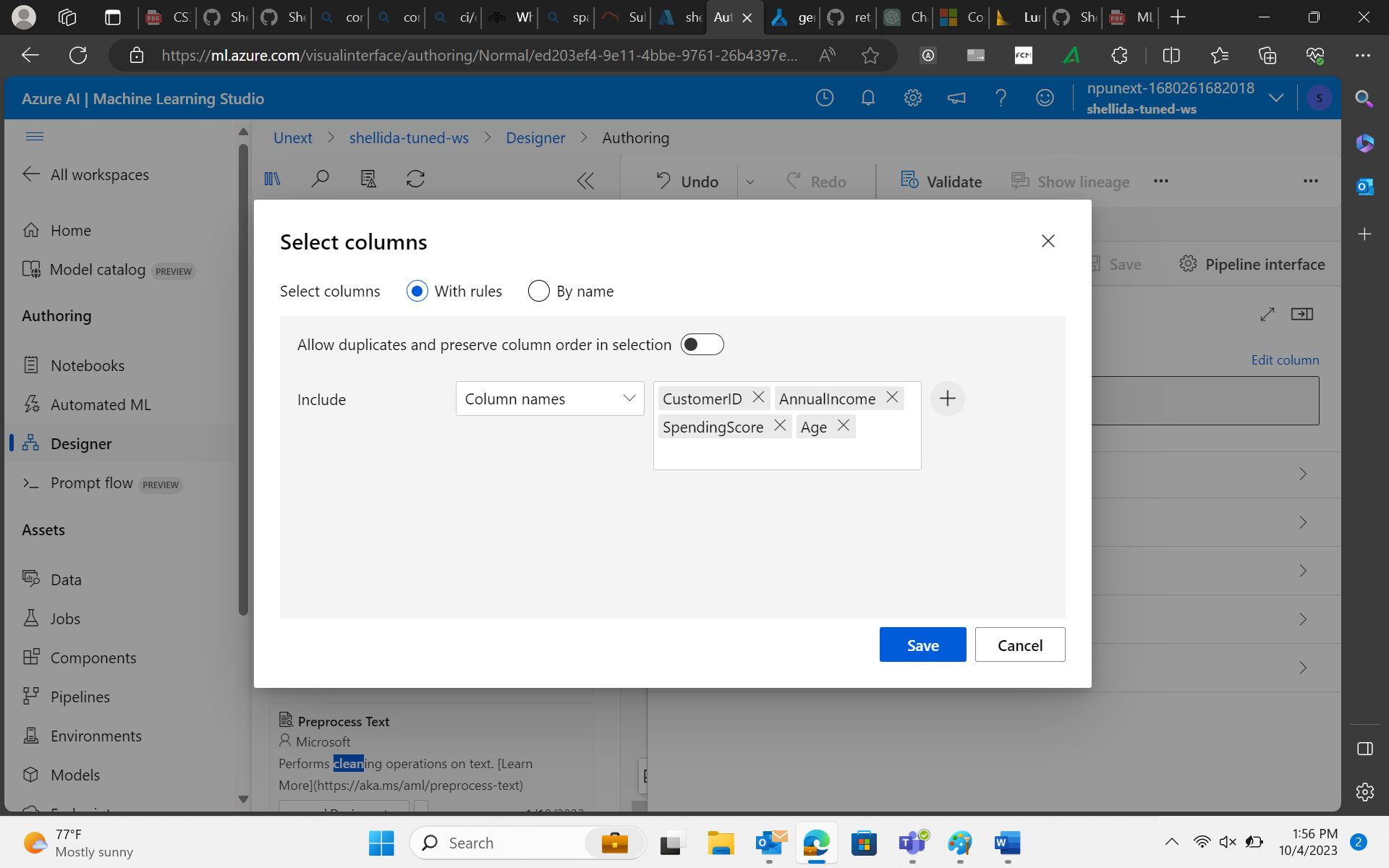


1. Drag and drop customer-data\_analysis.

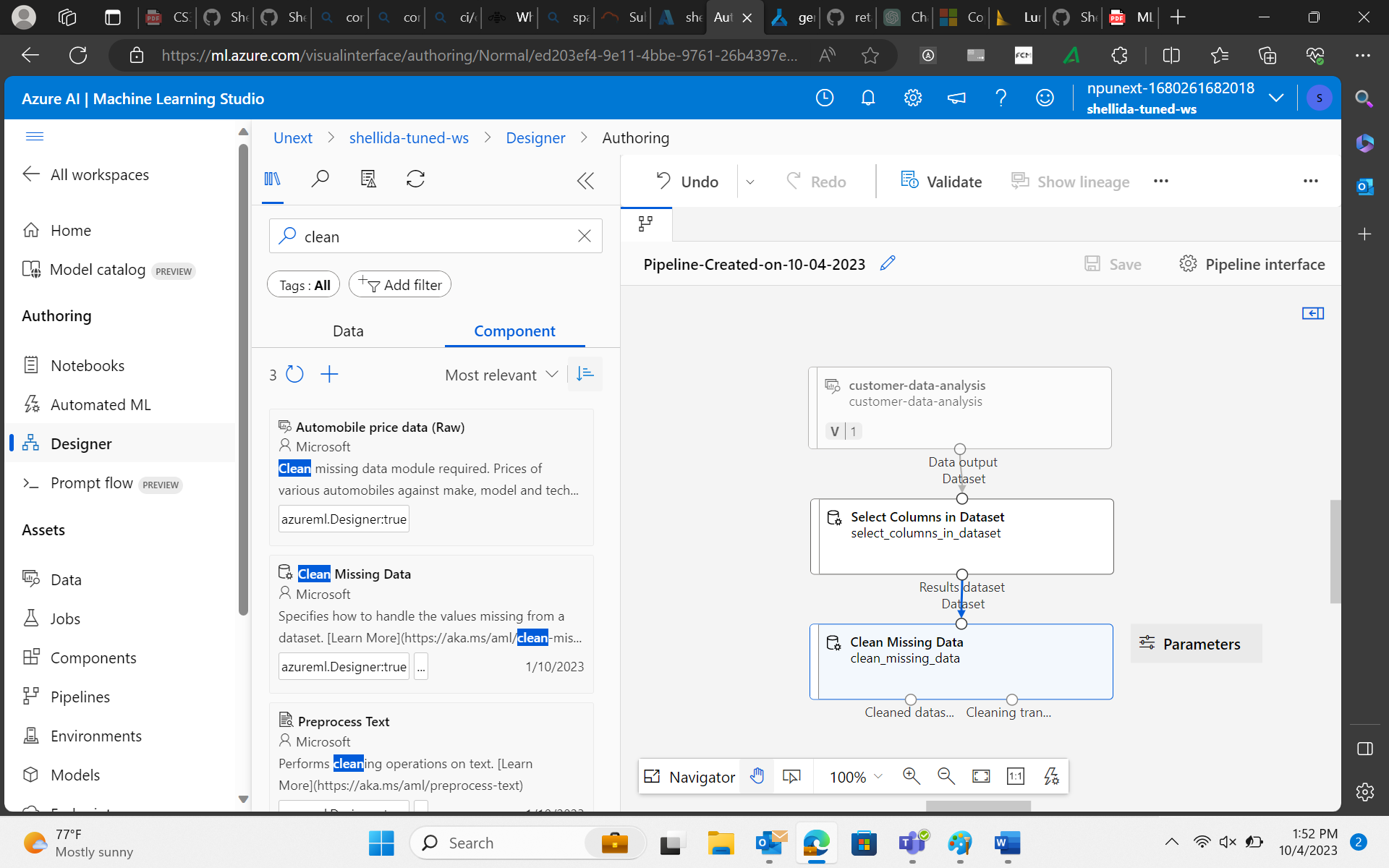


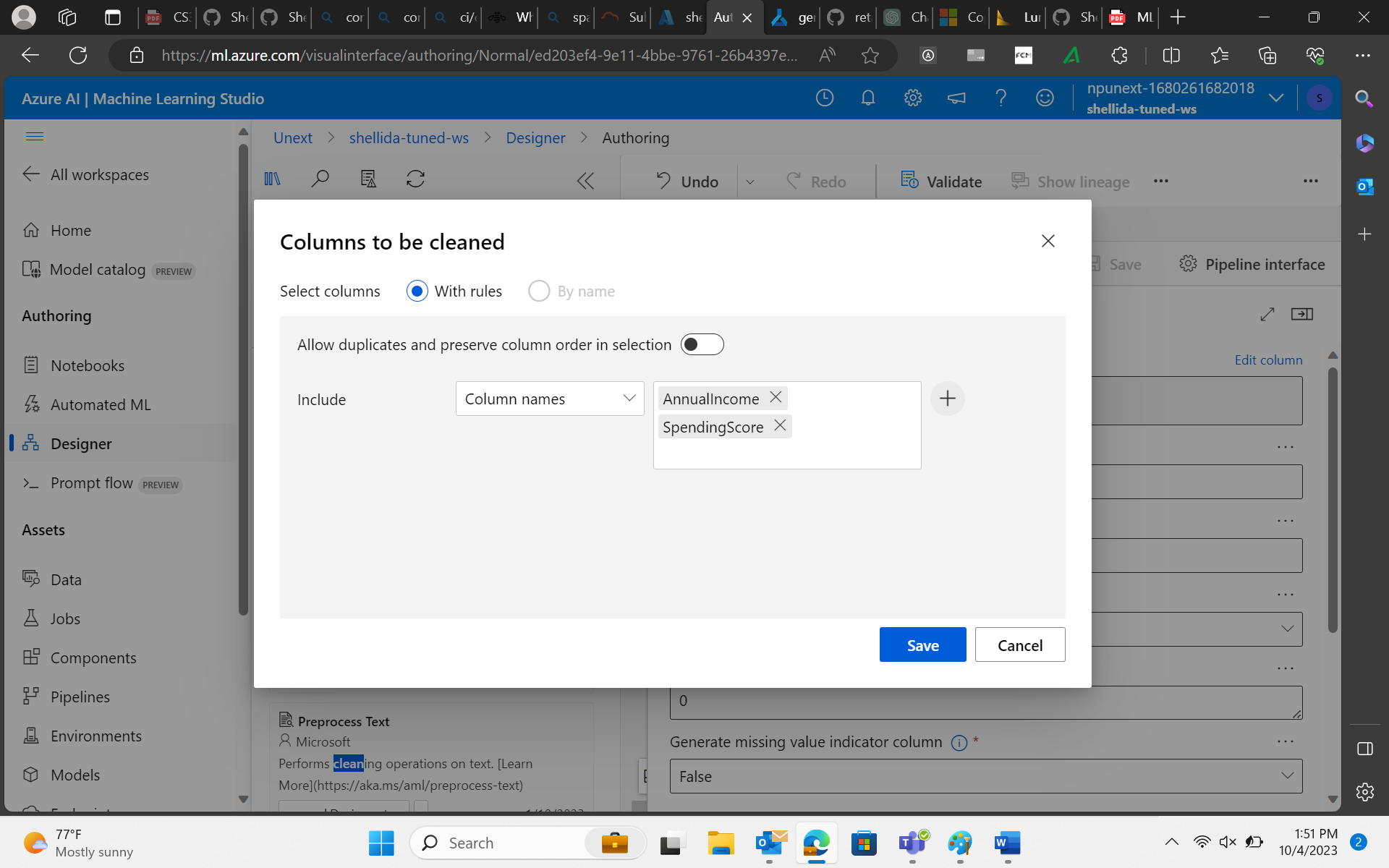
1. Perform Data Selection.

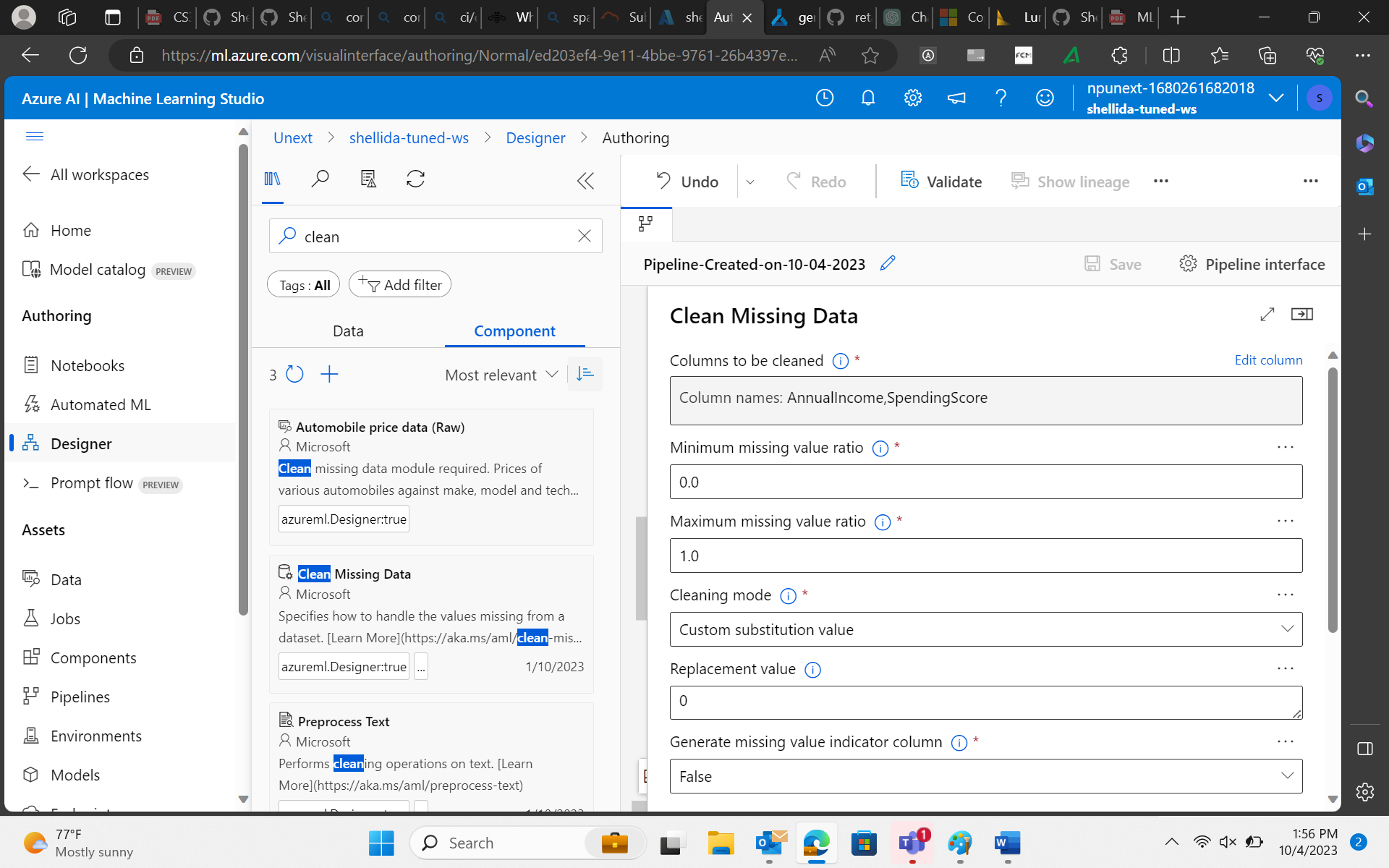


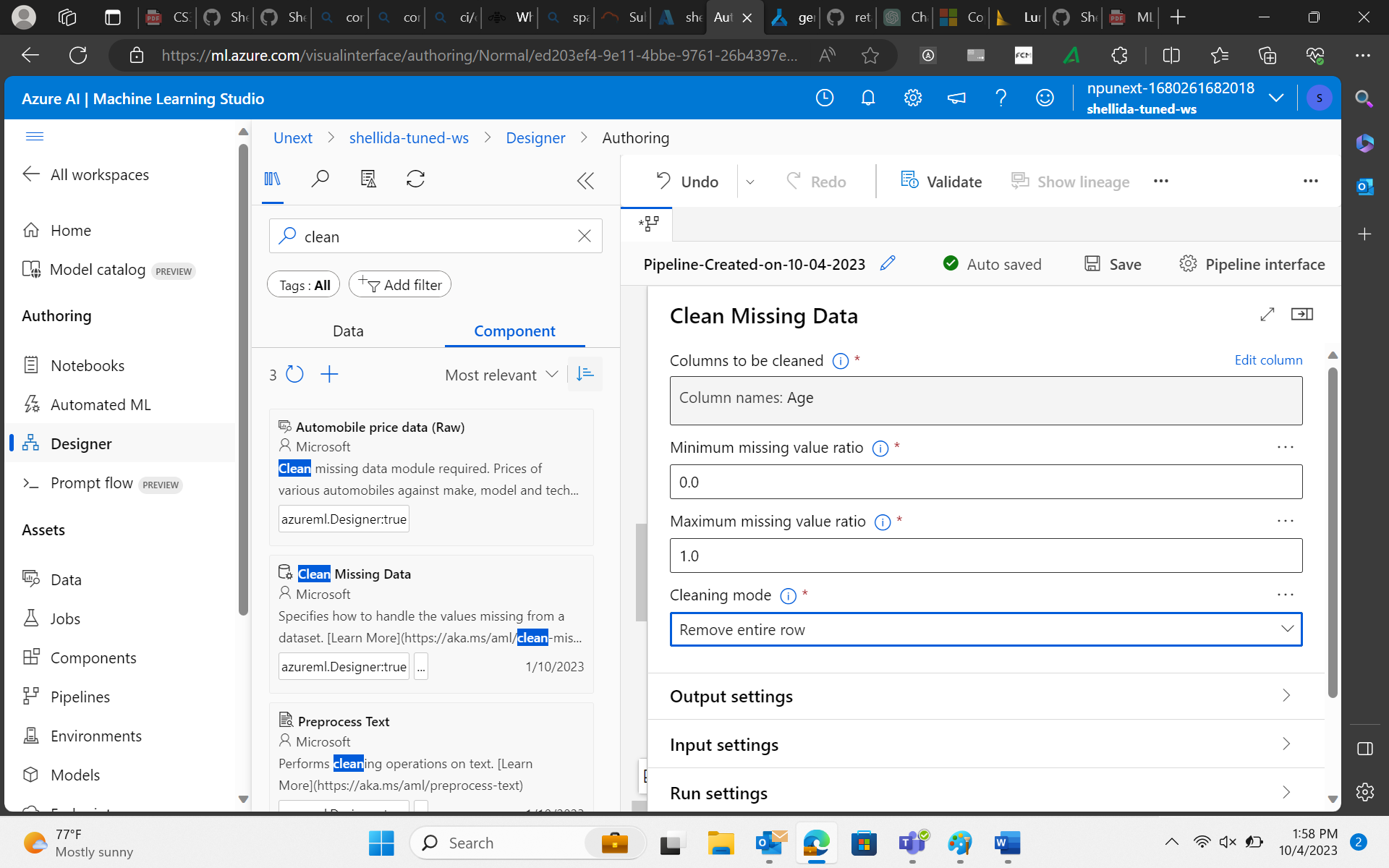


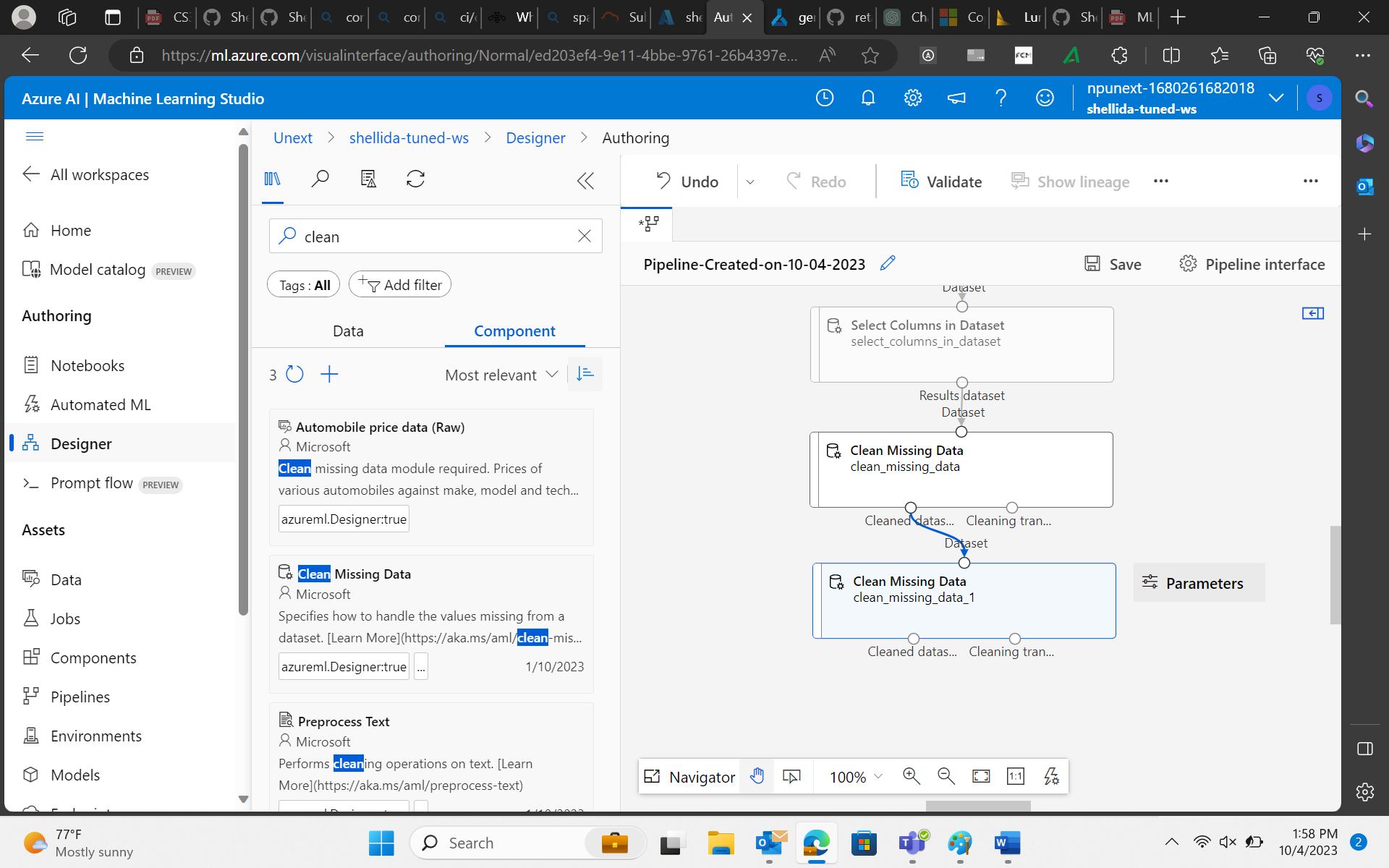
1. Perform Data Cleansing



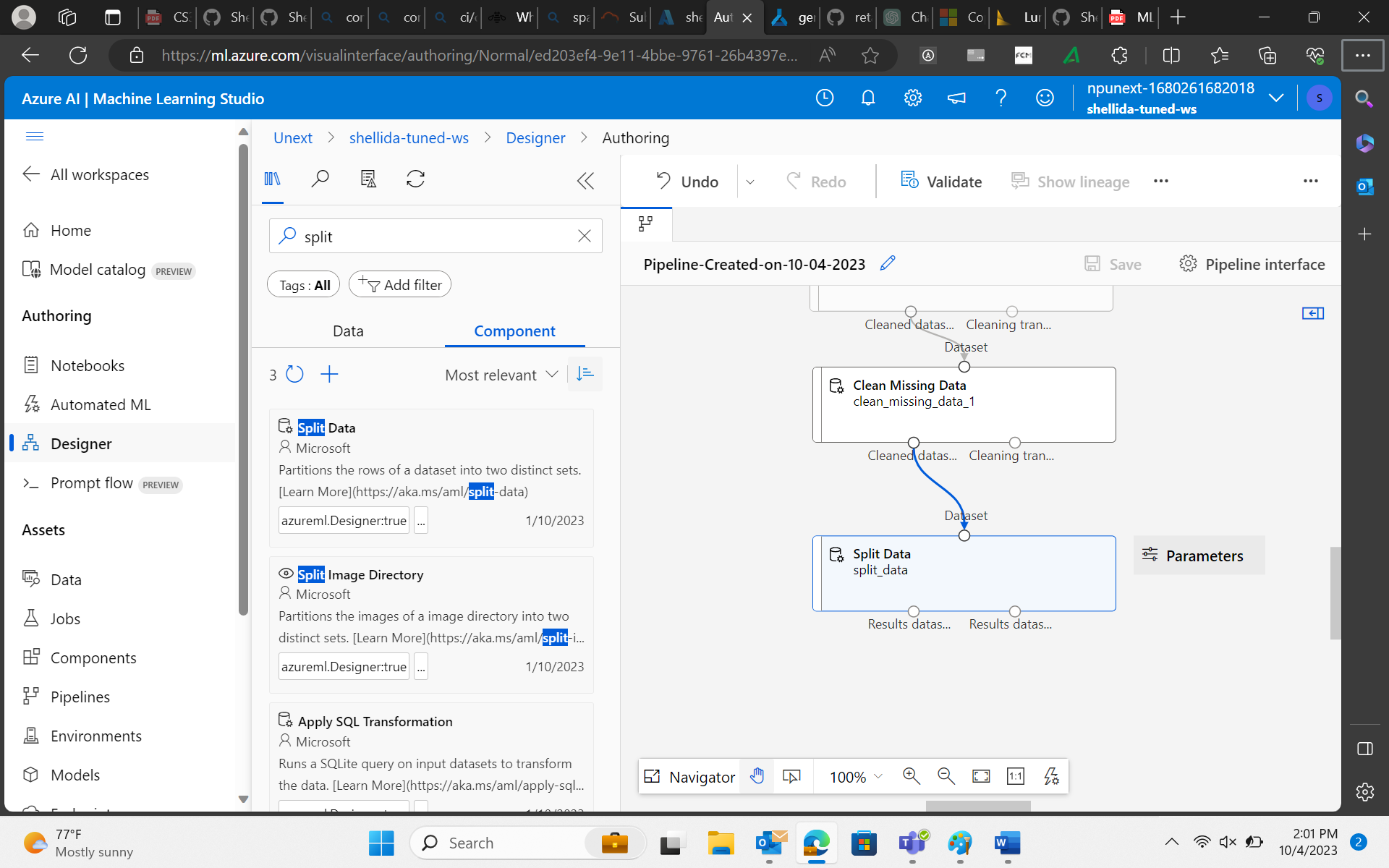


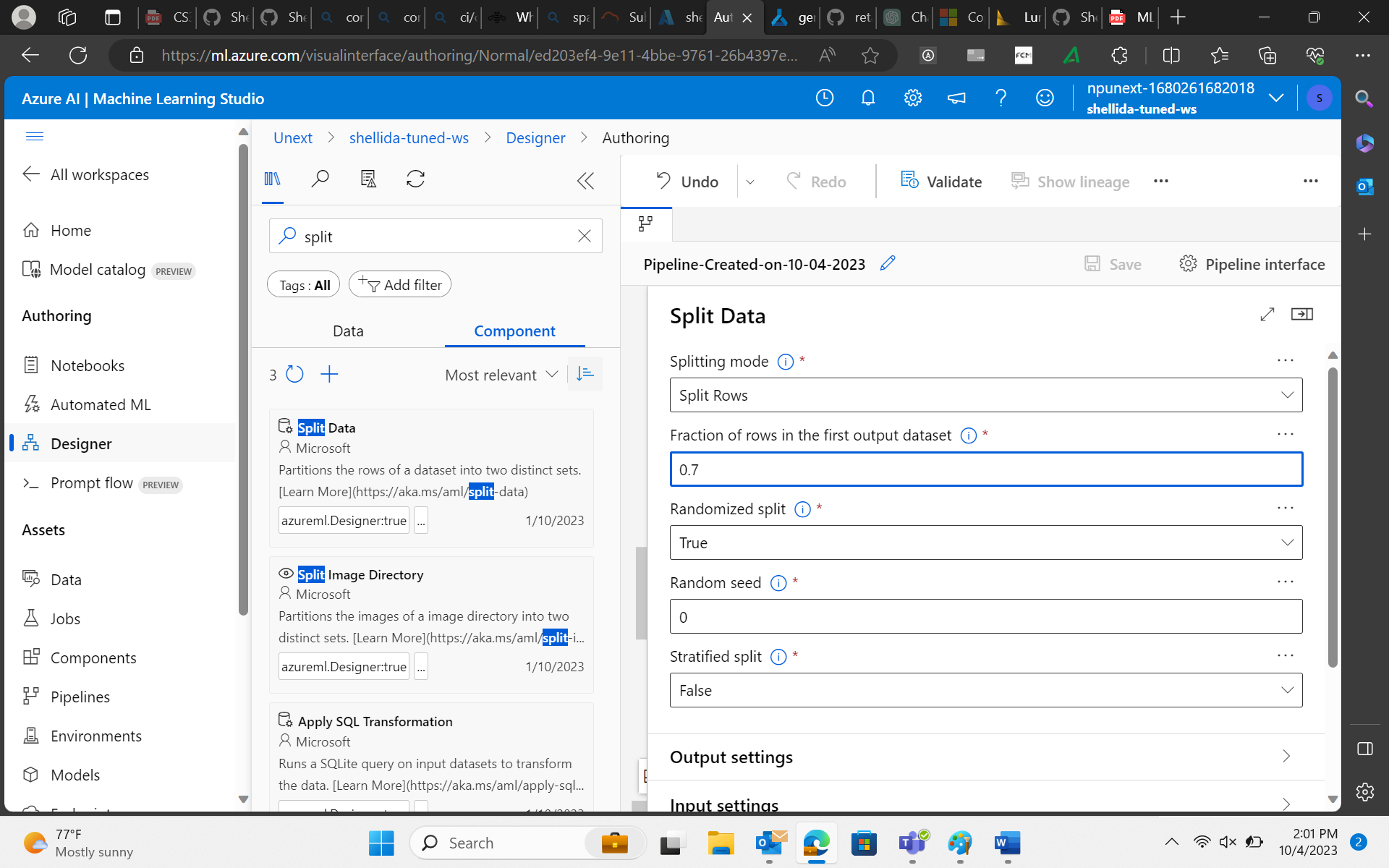




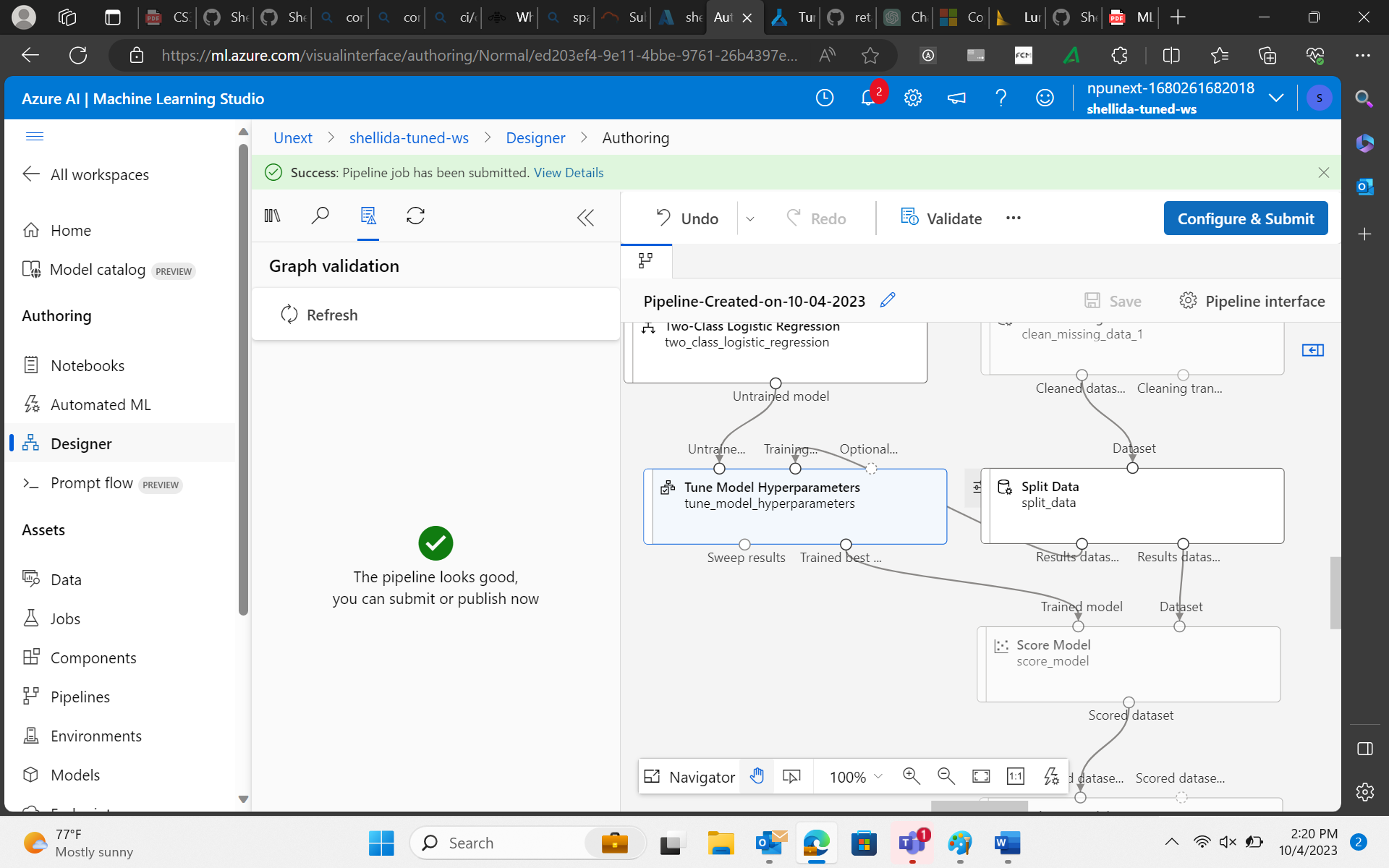


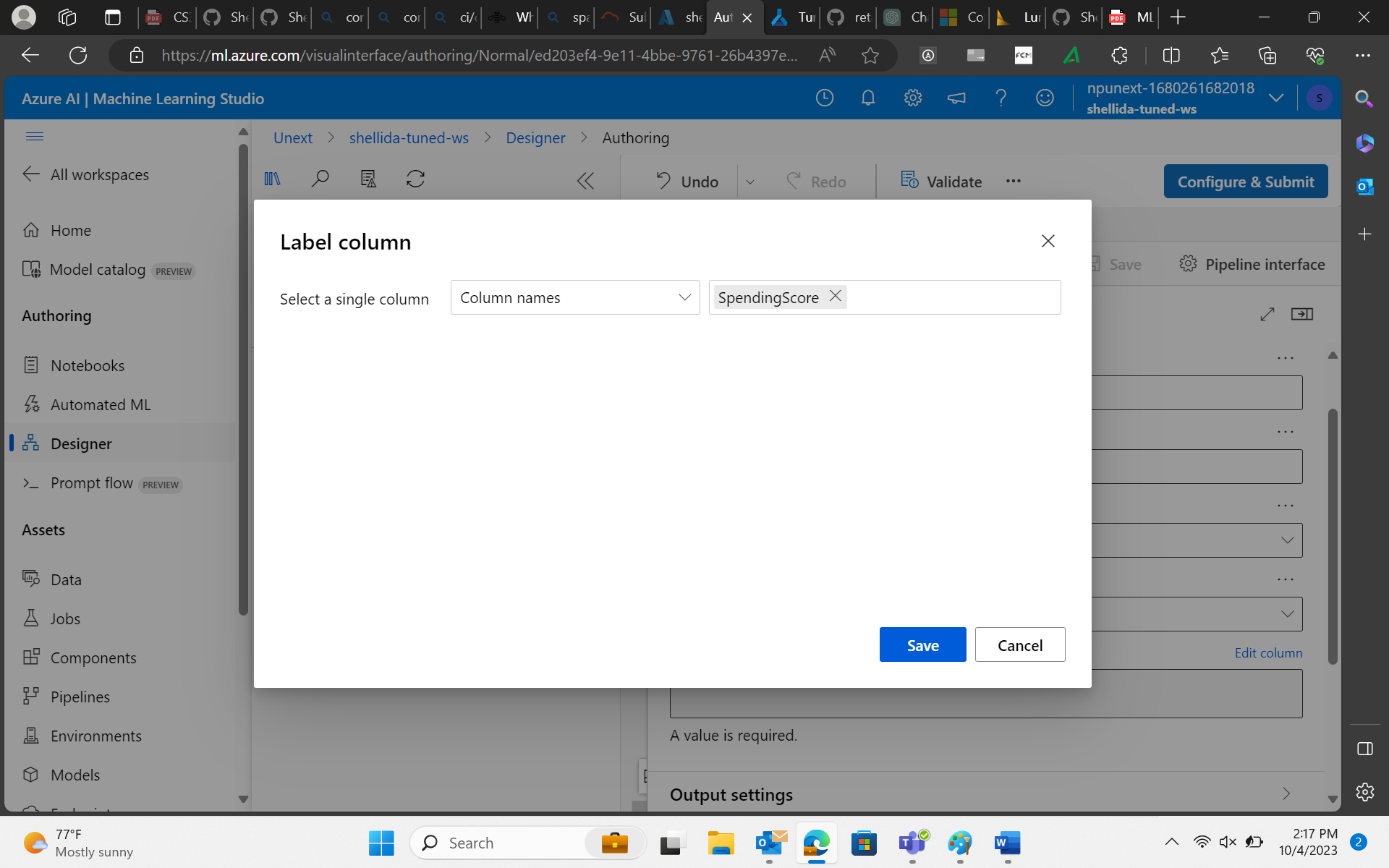
1. Split the data into training and testing datasets.



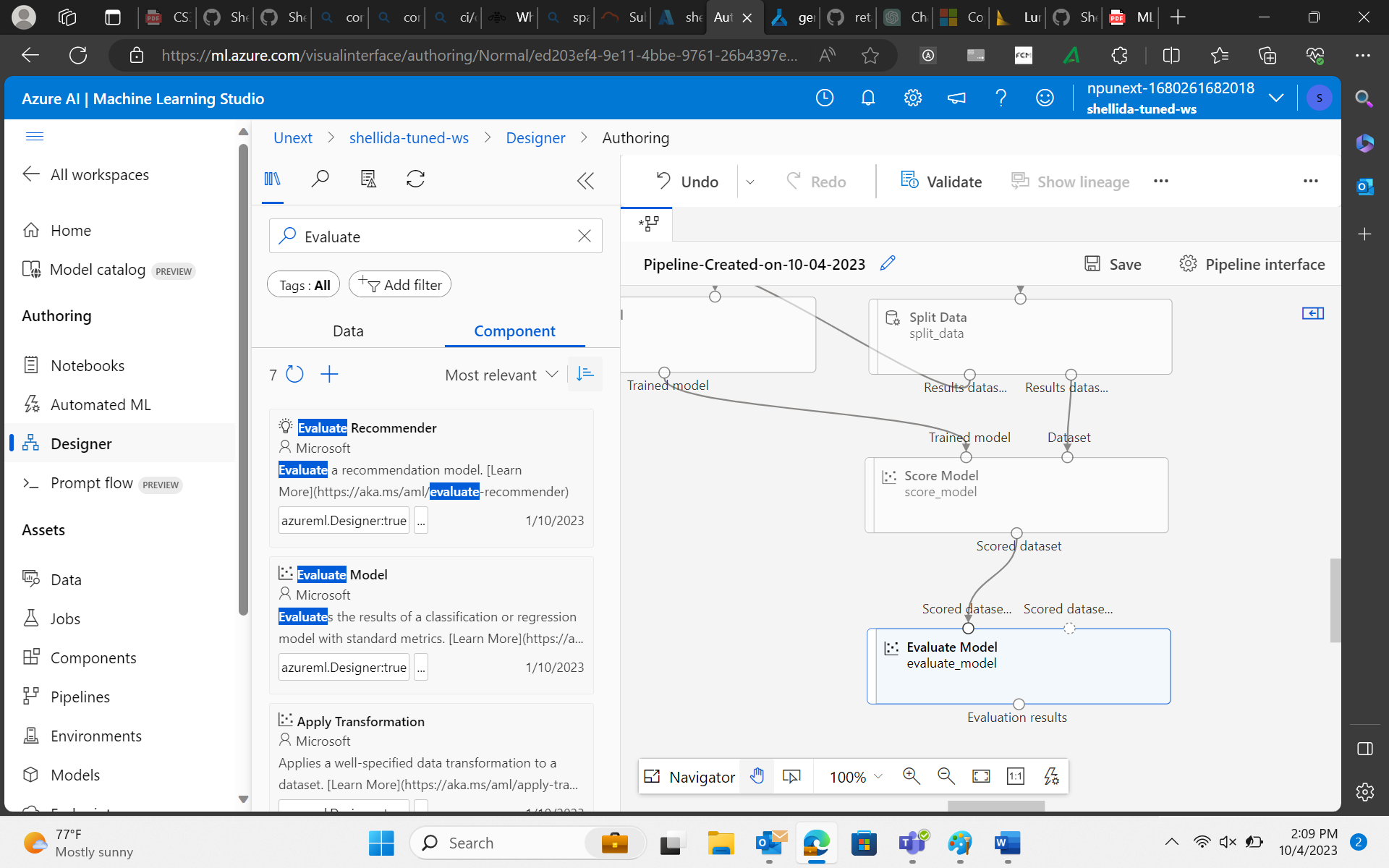


1. Tune the Model using Two class Logistic Regression.

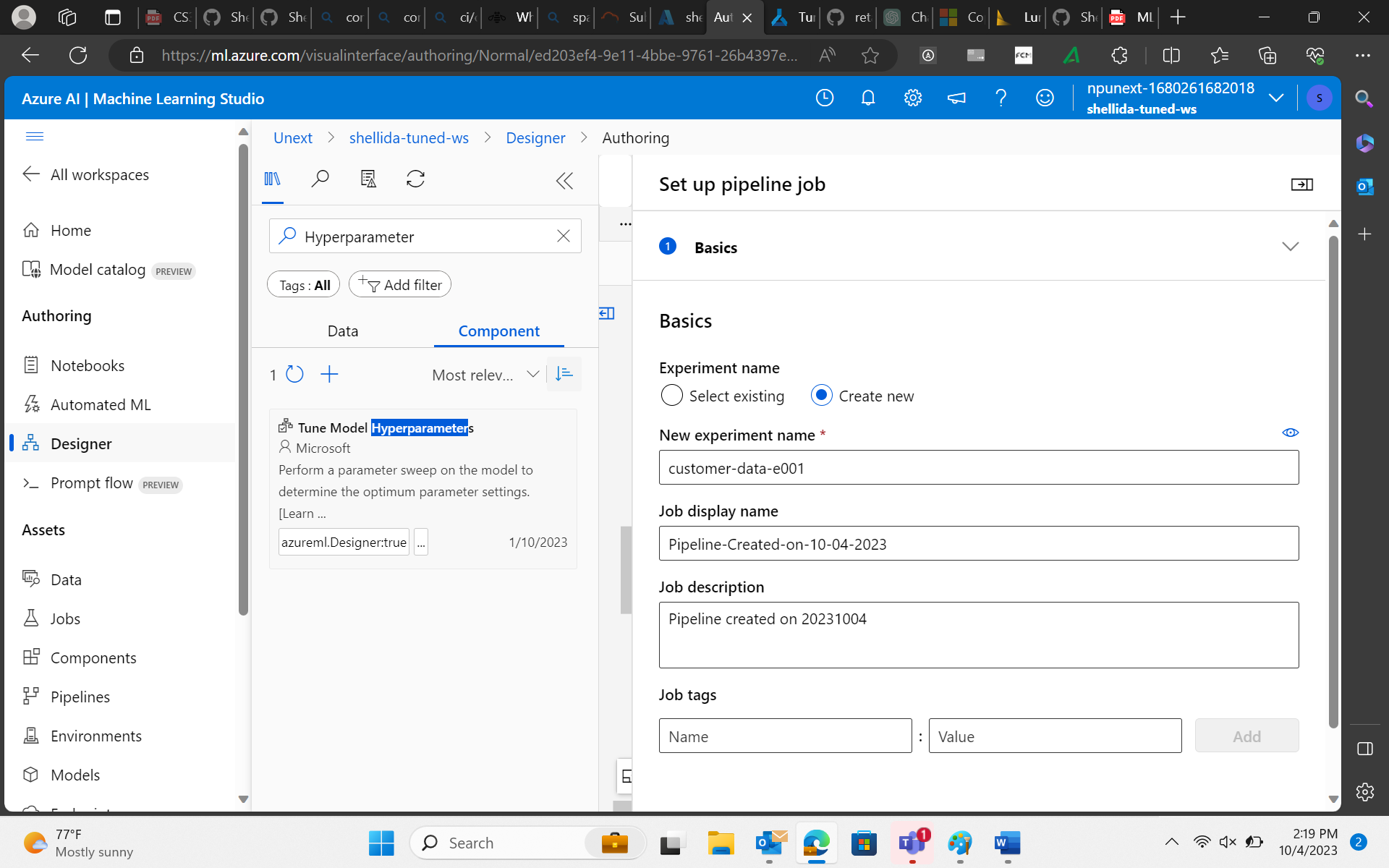


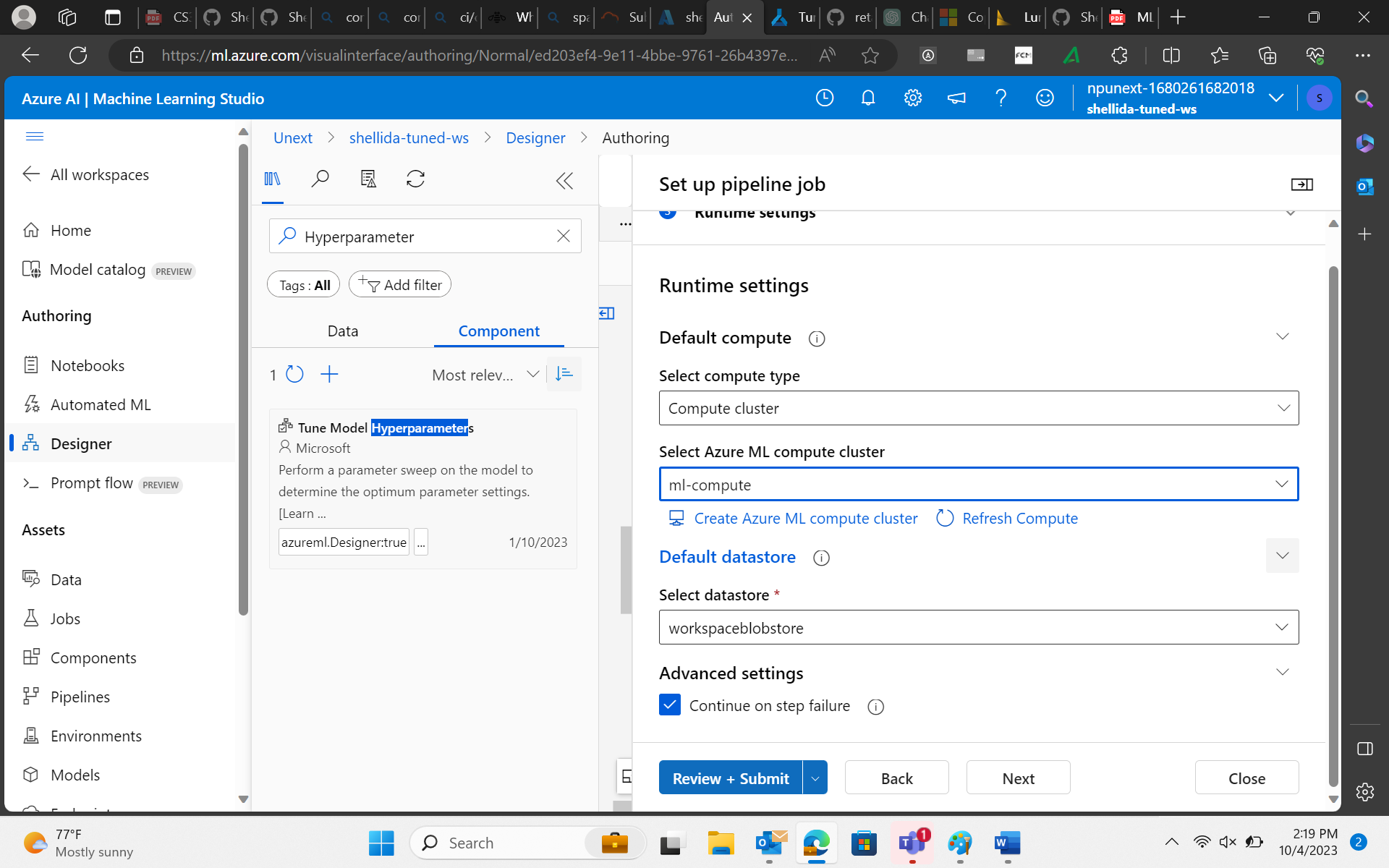


1. Score and Evaluate Model

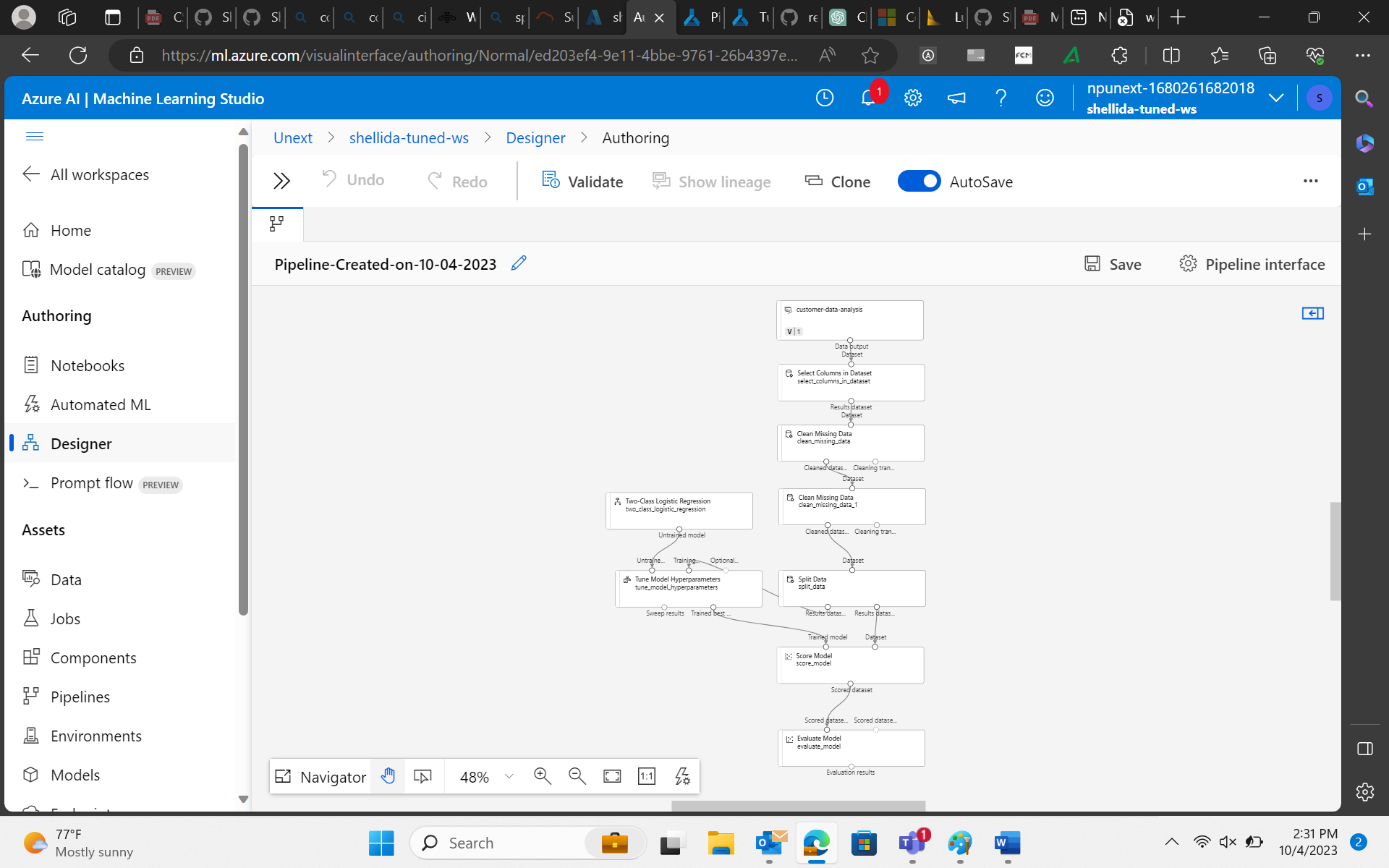


1. Configure and submit









**1.What are the key steps involved in preparing the dataset for training a machine learning model using Azure Machine Learning? Briefly explain each step**

Here are the key steps involved:

1. Data Collection: Collect and gather the data from various sources, such as databases, local files, APIs, or external datasets, Azure Storage.
2. Data Selection: Explore the dataset to gain a deep understanding of its characteristics, including the distribution of features, missing values, and outliers and select the required columns.
3. Data Cleaning: Address missing data by imputing values or removing rows/columns with missing values, depending on the impact on the dataset.
4. Data Transformation: Convert categorical variables into numerical representations through techniques.
5. Feature engineering: Create new features or modify existing ones to better represent the underlying patterns in the data.
6. Data Splitting: Split the dataset into training, validation, and test sets. Common ratios are 70-80% for training, 10-15% for validation, and 10-15% for testing.Randomly shuffle the data to ensure that each subset is representative of the overall dataset
7. Data Preprocessing Pipelines: Create data preprocessing pipelines to automate and streamline the data preparation process.
8. Data Versioning and Tracking: Use Azure Machine Learning's dataset versioning and tracking capabilities to keep records of changes to the dataset. This helps maintain a clear history of dataset modifications and ensures reproducibility
9. Data Validation: Validate the dataset in the Azure Machine Learning workspace to ensure that it is correctly formatted and suitable for training.
10. Data Integration with Model Training Pipeline: Integrate the prepared dataset into the machine learning model training pipeline within Azure Machine Learning. Define the input data source for your model training script or pipeline.

**2.Why is it important to split the dataset into training and testing sets when developing a machine learning model? How does this help in model evaluation?**

A. Splitting the dataset into training and testing sets is crucial for model development. It provides an independent dataset for evaluating the model's performance, measuring its ability to generalize to unseen data. Testing data helps identify overfitting, where a model memorizes the training data but fails to generalize, improving model reliability. Testing sets enable the calculation of various metrics (e.g., accuracy, precision, recall) to assess model quality objectively. Testing sets guide the fine-tuning of hyperparameters to optimize model performance. It simulates real-world scenarios where models encounter new, unseen data, ensuring practical applicability.

3. **Describe a machine learning algorithm suitable for predicting customer purchasing behaviour in the given scenario. Explain why you chose this algorithm.**

A. In the scenario of predicting customer purchasing behavior, a suitable machine learning algorithm is the Random Forest. Random Forest is an ensemble learning algorithm that combines the predictions of multiple decision trees. This ensemble approach often results in more robust and accurate predictions. It is effective for both classification (predicting whether a customer will purchase or not) and regression (predicting the amount of purchase) tasks. It can handle various types of features, including categorical and numerical. It provides feature importance scores, helping to identify which customer features (e.g., demographics, past purchase history) contribute most to purchase predictions. This insight can inform marketing strategies. It naturally mitigates overfitting, a common challenge in predictive modeling, by averaging the predictions of multiple trees. This improves the model's ability to generalize to new customers. It is less sensitive to outliers in the data, making it suitable for scenarios where customer behavior may vary significantly. It can handle large datasets and is relatively easy to tune for optimal performance. Overall, Random Forest is a versatile and powerful algorithm that can capture complex relationships in customer data, making it a suitable choice for predicting customer purchasing behavior in a practical and effective manner.

4. **What is hyperparameter tuning, and why is it important in machine learning? Explain a technique used for hyperparameter tuning and its benefits.**

A. Hyperparameter tuning, also known as hyperparameter optimization, is the process of systematically searching for the best combination of hyperparameters for a machine learning model. Hyperparameters are configuration settings that are not learned from the data but are set prior to training. They significantly impact a model's performance and behavior. Hyperparameter tuning is important in machine learning for several reasons:

1. It aims to find hyperparameters that maximize a model's predictive accuracy, minimizing errors, and improving overall performance.

2. Proper hyperparameter values can help prevent overfitting (model memorizing the training data) or underfitting (model failing to capture patterns), resulting in a model that generalizes well to new data.

3. Tuning can make a model more robust to variations in the data, ensuring it performs consistently on different datasets or under different conditions.

4. Optimized hyperparameters can reduce training time and memory requirements, making the model more efficient.

One common technique for hyperparameter tuning is Grid Search: Grid Search: Grid Search involves defining a grid of possible hyperparameter values or ranges for a model. The algorithm systematically tests all possible combinations of hyperparameters within the predefined grid. For each combination, it evaluates the model's performance using cross-validation or a validation set. The best-performing set of hyperparameters is selected as the final configuration for the model.

Grid Search explores a wide range of hyperparameter combinations, ensuring that the best configuration is found within the specified grid. It uses a rigorous evaluation process, often with cross-validation, to assess model performance, reducing the risk of overfitting to the validation set. Simplicity: Grid Search is straightforward to implement and understand, making it accessible to both beginners and experienced machine learning practitioners.