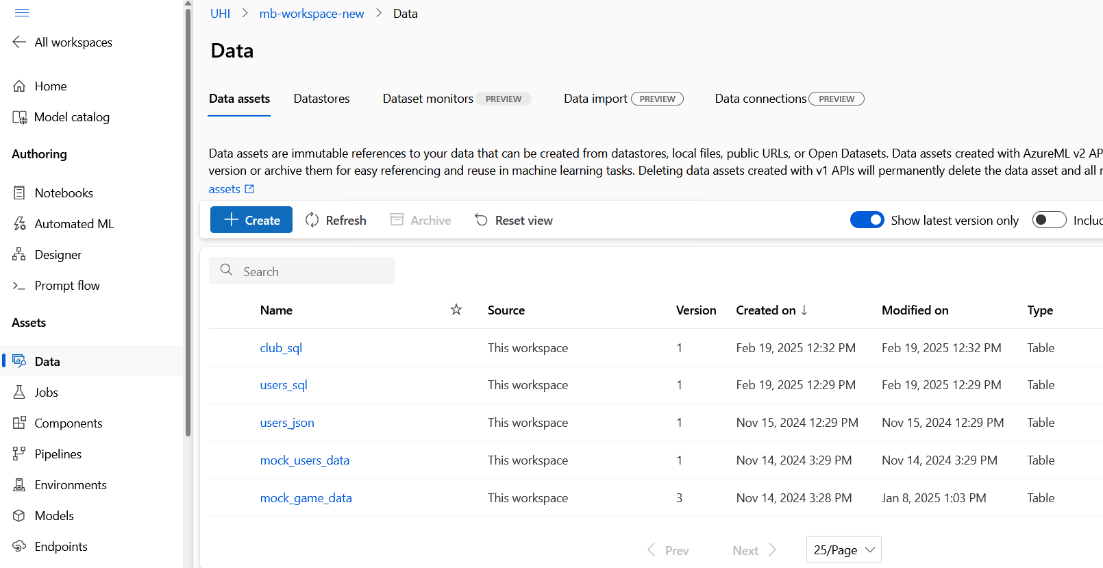
**MG Azure ML Pipeline Docs**

in Azure Portal, search resources (top search bar) for Azure Machine Learning

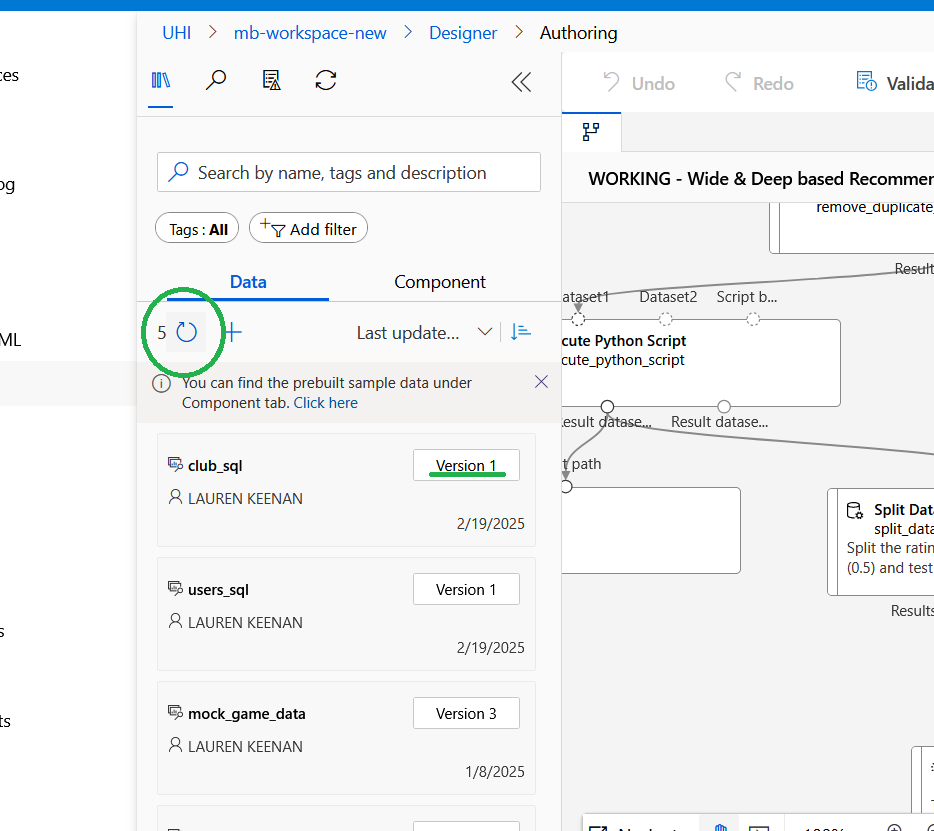
in Azure Machine Learning, click on `mb-workspace-new`

To access the ML pipeline, click on `Designer` tab on the left menu and select `WORKING - Wide & Deep based Recommender` from Pipeline drafts.

In the `Data` tab on the left menu are pipeline components manually created for the data files and tables in the account we want to use for ML.



You can create and update your components (but to my knowledge cannot delete them, only archive them), so it's good practice to simply update a component, refresh the components in the `Designer` and click and drag the newest version into the designer to use.



## Data sources used in Pipeline (at the time of writing):

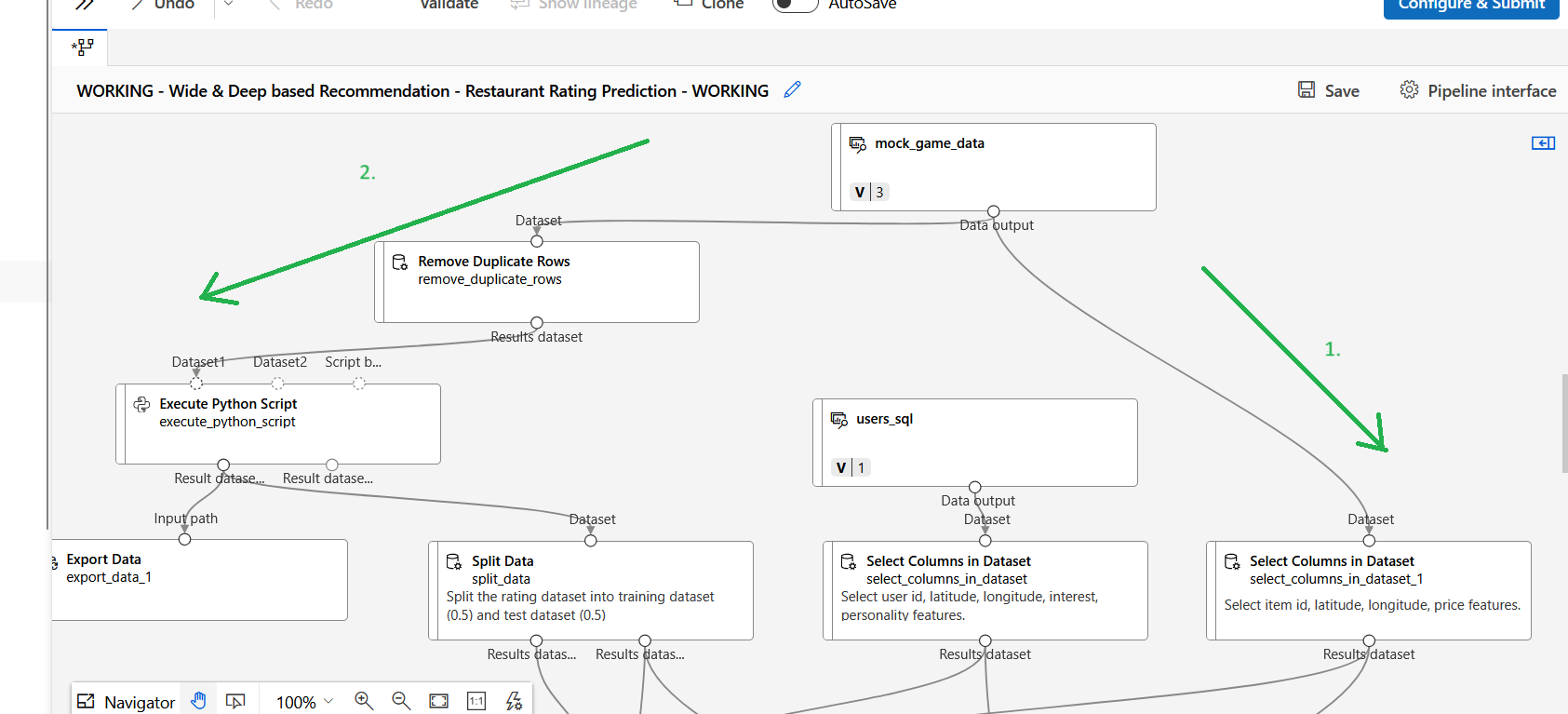
- `mock\_game\_data` = `MOCK\_GAME\_DATA2.jsonl` in Blob storage

- `users\_sql` = `dbo.users\_data` SQL table

`mock\_game\_data` currently includes the entire club JSON object in each game and it can be read as a column by the ML in `select\_columns\_in\_dataset\_1`

## How it works

Data from `mock\_games\_data` is used in `select\_columns\_in\_dataset\_1` to select features of the game to feed directly into the `Score Wide and Deep Recommender` component (Arrow 1). It simultaneously is pre-processed into a dataset of userId, gameId, ratings (Arrow 2). This is then split into training and test datasets.



The training dataset is passed into `Train` and the other is fed into `Score` and `Evaluate` as a test set to be compared with the eventual Scored dataset produced by the other data processed in the `Score` component.

`users\_sql` has user features selected as columns fed into both the `Train` and `Score` components.

All together the ML model uses features from `users\_sql` and `mock\_game\_data`, and userId, gameId, and ratings extracted from `mock\_game\_data`

Finally, the `Export Data` component attached to `Score` outputs the pipeline results to `MLresults` in the `mbmlstorage` Blob storage account. It should save as a csv and it's parsed as a csv from the frontend app, but it shows as either JSONL or unknown file type. Works though.

Each pipeline component can be tweaked by double clicking and exploring the settings, but it's not advised to edit Output and Input settings unless you know what you're doing. (an error I encountered was an entire new resource group being made which exceeded the free tier and charged me)

The `Score` component can be edited to change the recommendation type between Item Recommendation/content based (currently) or Ratings Prediction. The frontend application will require different postprocessing for each.

As data grows and becomes more rich, more/different columns can be selected for use in the ML by editing the `Select Columns` components

Remember to save any changes at the top right tab.

## Run the model

Navigate back into `mb-workspace-new` and scroll to the bottom of the screen. Select `Add compute instance`, you can just choose the cheapest GPU (usually around $0.10 per hour). When it's up and running navigate back to the `Designer` and click Configure and Submit in the top right.

This will run the pipeline. A Green snackbar should appear at the top left of the Designer, click on it to see the job in process.

The data input and output of this pipeline interact with the frontend app either directly (Blob) or via a custom API (SQL). Changes made by users in the app reflect in the ML Data sources and the ML output is immediately accessible to the frontend app. This Pipeline is manually run and used as a cloud web service.

If/when the project scales it could be deployed per Azure documentation, but it so far seems costly and resource intensive.

***\*\*Remember to delete the compute instance when not using it, or you’ll be charged\*\****