**AI MACHINE TEAM**

**Done by:**

Anirudh Gokamalla

Himasahana Remella

Srivardhan Chavatapalli

**Abstract:**

This project aims to develop a robust image classification system utilizing Azure Machine Learning services to automatically categorize images into predefined classes. By leveraging Azure ML's advanced computational capabilities and scalable infrastructure, we propose a system that can handle large datasets with high efficiency and accuracy. This endeavor not only contributes to the field of computer vision but also demonstrates the practicality and efficiency of cloud-based machine learning solutions for complex tasks such as image classification. The expected outcome is a model that can classify images with high precision and recall, which will be evaluated using standard metrics such as accuracy.

**Architecture:**  
 A diagram of a work space

Description automatically generated

Toelaborate on the architecture for image classification using the MNIST dataset in Azure ML Studio, let's break down each component in detail:

1. The Data Ingestion:

* The process starts with importing the MNIST dataset into Azure ML Studio.
* MNIST is a widely used dataset for handwritten digit recognition, consisting of 60,000 training images and 10,000 test images.
* This step involves accessing the dataset, which is commonly available in a preformatted structure suitable for machine learning tasks.

1. Data Preparation:

* This stage involves preprocessing the images to make them suitable for training a neural network.
* Common steps include normalizing the pixel values (scaling them to a range, often 0 to 1), and reshaping the images into a format compatible with the input layer of the neural network.
* This may also include dividing the dataset into training and validation sets for more effective model training.

1. Model Training:

* In this phase, a model is trained on the MNIST data.
* Azure ML Studio offers various tools and frameworks to design and train the model.
* The training process involves feeding the preprocessed images to the network and adjusting the model parameters (weights and biases) using algorithms like backpropagation, based on a loss function.

1. Model Evaluation:

* After training, the model's performance is evaluated using metrics such as accuracy, precision.
* This typically involves testing the model on a separate set of data (test dataset) that it hasn't seen during training.
* Evaluation helps in understanding how well the model will perform in real-world scenarios or with new, unseen data.

1. Deployment:

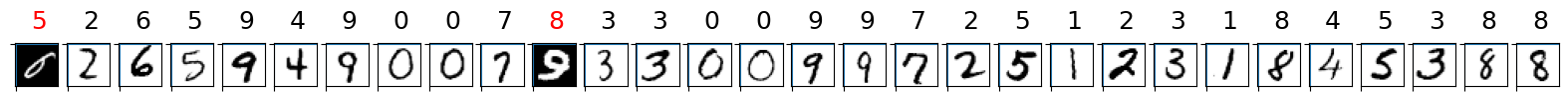
* The final step is deploying the trained model for inference.
* In Azure ML Studio, this can involve setting up a cloud service where the model is hosted.
* Users can then input new data (handwritten digits) into the service, and the model will provide predictions (digit classifications).

**Features:**  
Our Model:

* Accepts user inputs for handwritten numbers dataset.
* Processes inputs using image model and predicts the image.
* Dataset loads in the pipeline and processes with the data given.
* Output is given with a confusion matrix and data is trained and input data is tested.
* Delivers an image of the digits with the accuracy of 0.91.

Example Session:

**User:** MNIST dataset

**Model**: 

**How it Works:**

1. Data Management

* How it Works: Azure ML Studio allows users to upload or connect to datasets stored in various formats and locations (like Azure Blob Storage). For the MNIST dataset, it's often in a pre-processed, ready-to-use format.
* Data Visualization: Tools within the studio enable quick exploration and visualization of the data, helping to understand its distribution, range, and other characteristics.

1. Preprocessing and Data Transformation

* Normalization and Reshaping: Using the studio's built-in tools, each image in the MNIST dataset (originally grayscale images of 28x28 pixels) can be normalized (scaling pixel values to a range of 0 to 1) and reshaped if necessary (flattening the 2D array into a 1D array for certain types of neural networks).
* Dataset Splitting: The platform facilitates the splitting of data into training, validation, and test sets, a crucial step for effective model training and evaluation.

1. Model Building and Training

* Azure ML Studio offers a user-friendly interface to build image classification models. You can select different layers and set their parameters (like the number of neurons in a dense layer, activation functions, etc.).
* Training Process: Once the model is designed, you can train it using the MNIST data. The platform allows you to set various training parameters, like the number of epochs, learning rate, and batch size.

1. Hyperparameter Tuning and Optimization

* Azure ML Studio's feature can automatically test various hyperparameters to find the most effective ones for your model.
* Customizable Settings: For advanced users, there's the flexibility to manually adjust and experiment with different hyperparameters.

1. Model Evaluation and Validation

* Performance Metrics: Post-training, the model's performance is evaluated using metrics such as accuracy, precision etc. Azure ML Studio provides these metrics out of the box.
* Error Analysis: Tools like confusion matrices help in understanding the types of errors the model is making, which is crucial for further refinement.

1. Deployment and Model Serving

* Web Service Deployment: The trained model can be deployed as a web service on Azure, making it accessible for inference via HTTP requests.
* Scalability: Azure’s infrastructure allows the service to scale based on demand, ensuring consistent performance even with high loads.

1. Monitoring and Maintenance

* Performance Monitoring: Once deployed, Azure ML Studio offers tools to monitor the model's performance in real-world scenarios, tracking metrics like throughput, response time, and accuracy.
* Model Updating: Based on the performance data, you can retrain your model with new data or adjust it to maintain or improve its accuracy over time.

**Try it Yourself:**  
To gain hands-on experience with Azure ML Studio using the MNIST dataset for image classification, follow this step-by-step guide. This exercise will help you understand the workflow and functionalities of Azure ML Studio.

**Step 1: Set Up Your Environment**

Log in to your Azure portal. Navigate to Azure Machine Learning Studio and create a new workspace if you haven't already.

**Step 2: Data Ingestion**

In Azure ML Studio, find the option to import datasets. Import the MNIST dataset, which is usually available in Azure ML Studio as a sample dataset.

**Step 3: Explore and Prepare the Data**

* Use the data visualization tools to explore the MNIST dataset.
* Normalize the images and reshape them as required for your neural network model.
* Split the dataset into training, validation, and testing sets.

**Step 4: Train the Model**

For the code part of training the model and deploy I have a sample code it will be easier if you can unzip the folder and upload it in your notebook.

**Step 5: Evaluate Your Model**

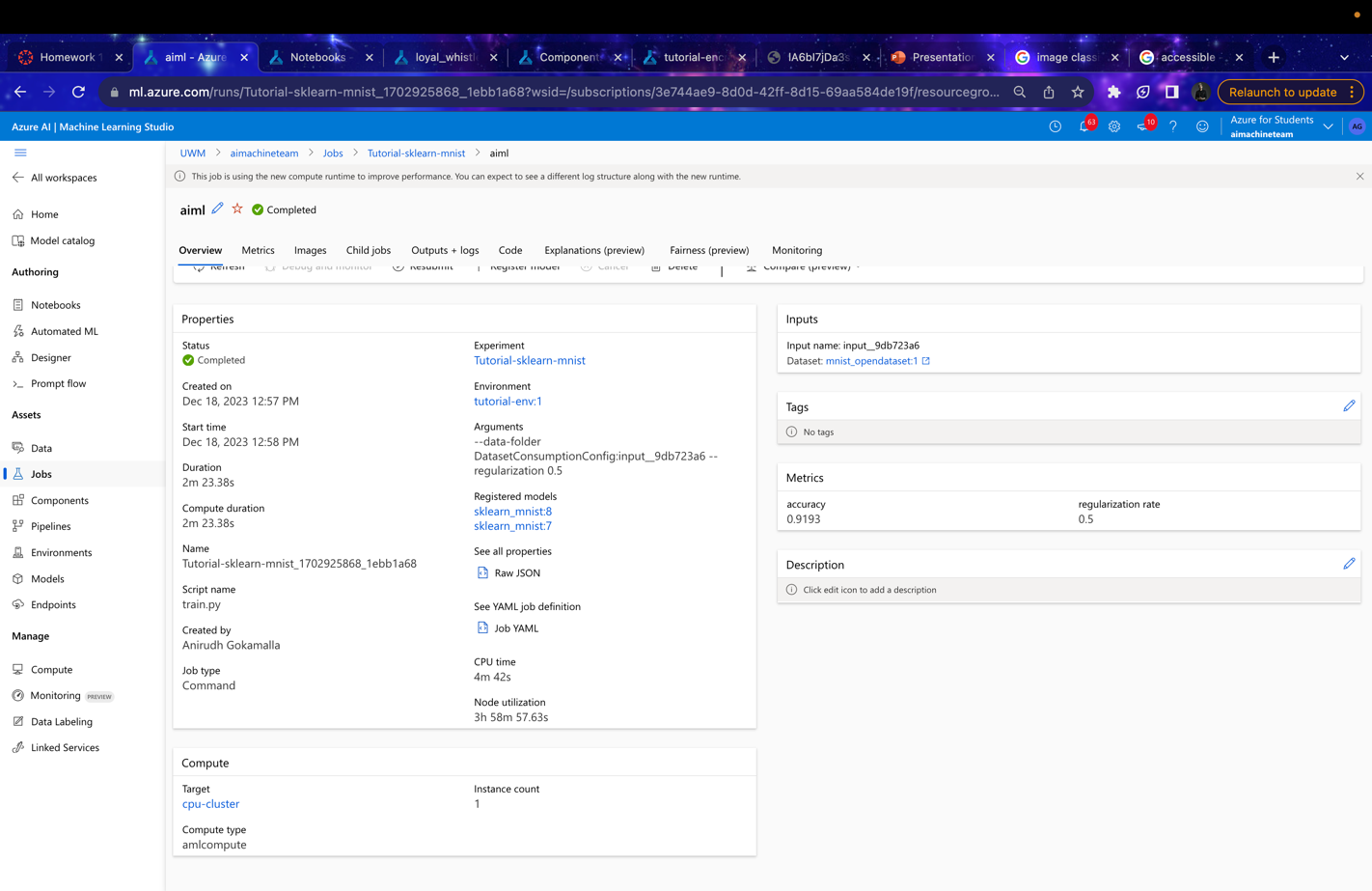
After training, evaluate the model using the test set. Check the accuracy and other metrics to assess the model’s performance. and for compute you can buy the azure ML python 3.8 compute which cost 0.21$/hr.



**Step 6: Register the Model**

Once satisfied with the model's performance, deploy it as a web service.

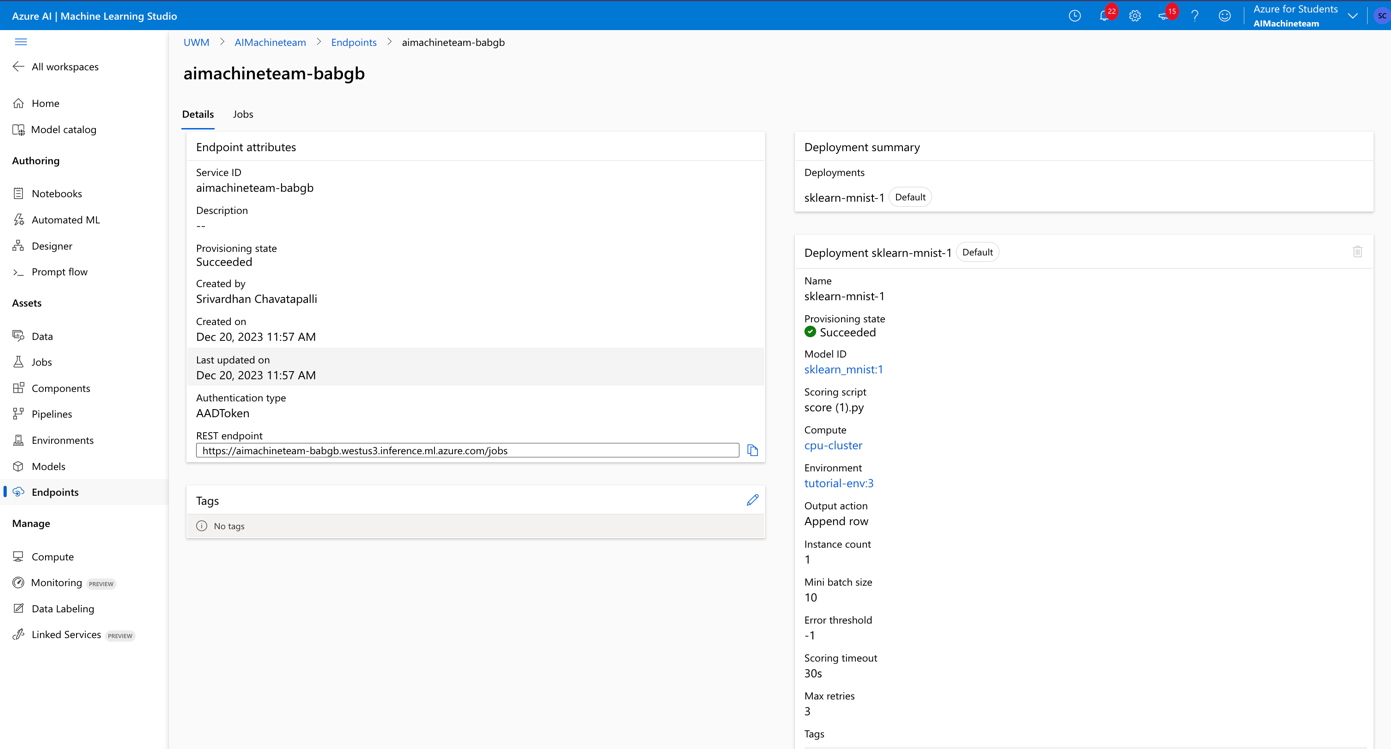
Azure ML Studio will guide you through the register process. And it will look like this after completion

****

**Step 7: Deploy the Model**

Once satisfied with the model's performance, deploy it as a web service.

Azure ML Studio will guide you through the deployment process, including setting up an endpoint for your model



**Step 8: Test the Deployed Model**

Use the provided endpoint to send requests to your model with test images. Check the responses to ensure that the model is classifying the digits correctly. But after the deployment am getting error in the YAML file and after trying to correct it as this machine learning products are new to even if am getting the RESTAPI endpoint my subscription got finished and I will work on it future but overall, the work is very helpful for handwritten documents and images.

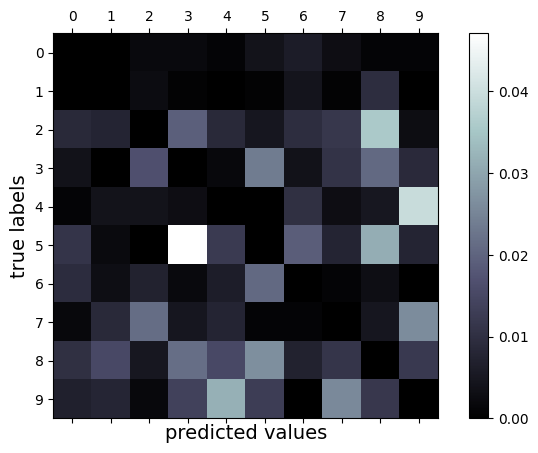
**Step 9: Monitor and Maintain**

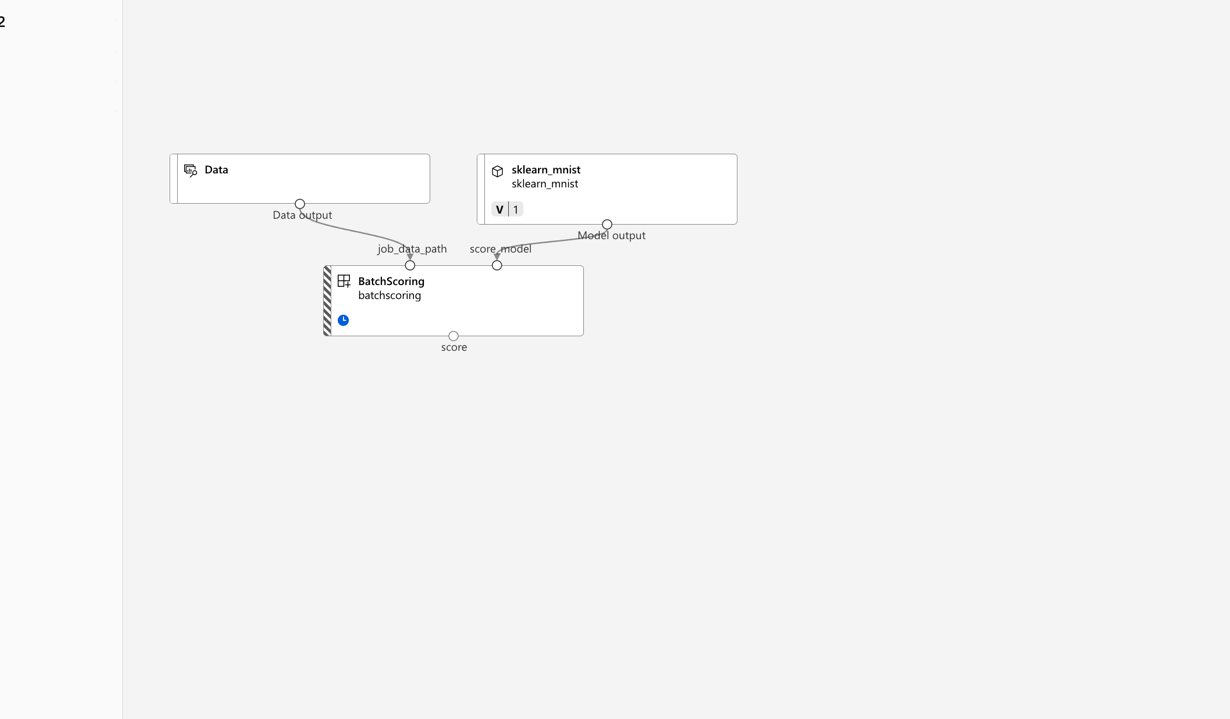
Set up monitoring tools to track the performance of your deployed model.

Plan for periodic updates and retraining with new data to keep the model accurate.

and there will be a confusion matrix with which you can analyze and overall, it has accuracy of 0.91 which is quite good and also using with the batch endpoint you can do it on pipeline.

In this graph, the X axis represents the actual values, and the Y axis represents the predicted values. The color in each grid represents the error rate. The lighter the color, the higher the error rate is. For example, many 5's are mis-classified as 3's. Hence you see a bright grid at (5,3).





**Benefits:**

The image classification project utilizing the MNIST dataset in Azure ML Studio lays the groundwork for diverse applications across sectors. In education, it could revolutionize interactive learning and automated grading by interpreting students' handwriting.

* Financial services might leverage it for cheque processing and automated data entry, enhancing accuracy and efficiency.
* Healthcare could see improvements in prescription interpretation and patient record management, while retail could benefit from better logistics through address verification. In security, signature verification and biometric authentication could become more robust.
* The arts could utilize it for font design and interactive installations, and robotics could advance in understanding handwritten instructions. Personal productivity tools like note digitization and handwritten search engines could become more sophisticated.
* Furthermore, government and public services could employ this technology for historical document preservation and streamlining processes like voter registration. This project's principles, though starting with simple digit recognition, have the potential to innovate and enhance operations across various industries, making everyday tasks more efficient and reliable.